

öS 60,-/sfr 7,-/Lit 7000/hfl 8,50 DM 7,–
AMIGA
Markt&Technik
12–'87
DAS COMPUTERMAGAZIN FÜR AMIGA-FANS
Grundlagen und Übersicht
Diskettenlaufwerke und Festplatten im Vergleich
RAM-Erweiterungen im Test
Speicheraufrüstung für den Amiga
Top-Listing zum Abtippen
Kopierprogramm mit tollen Extras
Neuer Grafik-Kurs
Vom Bit zum Bild
8MByte
4MByte
2MByte
1MByte
80MB
40MB
20MB
10MB
Das AMIGA-Magazin testet
arbdrucker unter 1000 Mark: MPS 1500
★ Brandaktuell: Vizawrite
★ Preiswert: Datamat



Von 0 auf 60 in 3 Sekunden

In Wahrheit haben wir untertrieben. Mit dem ANIM Programm in Videoscape 3D können bis zu 60 Bilder als flüssiger Bewegungsablauf in einer Sekunde abgespielt werden. Real time. Perfekt für die Videoproduktion am Schreibtisch. Perfekt für die Videopräsentation. Das Beste für Amiga.

Videoscape 3D wurde für alle Amigamodelle ab 512 K Arbeitsspeicher entwickelt. Das Programm verwendet räumliche Objekte mit verdeckten Kanten, diffuse Schatten einer Lichtquelle, reflektierende Spiegelungen und Drahtgitter-Modelle.

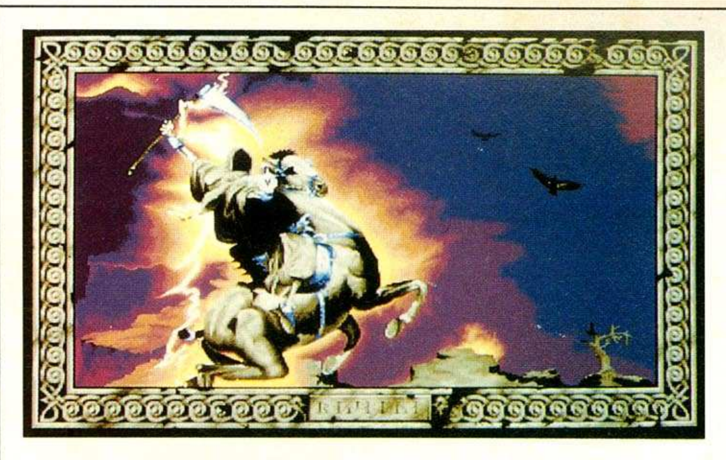
Mit Videoscape's Easy Geometrie Generator können Sie mit Leichtigkeit, geometrische Formen wie Würfel, Kegel, Quader und Kugeln erzeugen. Ferner können Sie mit dem visuellen Interface von Designer 3D ungewöhnliche Körper gestalten. Videoscape 3D beinhaltet sowohl eine Serie dreidimensionaler Objekte vom amerikanischen Computerfilmer Allen Hastings, als auch IFF Kunstwerke der Computergrafiker Jim Sachs und Richard LaBarre. Produzieren Sie einzelne Bilder und spielen Sie diese automatisch als Film ab, oder verwenden Sie den Effekt der Camera-Motion: Objekte oder ganze Landschaften scheinen auf den Betrachter zuzufahren. Videoscape 3D unterstützt alle gängigen Amiga Grafikauflösungen bis zu 704*440 Punkten, einschließlich Overscan und Interlace.

Videoscape 3D ist in einer deutschen Version mit PAL-Auflösung und deutschem Handbuch erhältlich. Zusammen mit anderen Produkten wie Aegis VideoTitler, Aegis Animator, Aegis Images oder Aegis Animation Workshop brechen Sie alle Rekorde.

Distributor: **softwareland**
Franklinstr. 27
CH-8050 Zürich
Schweiz

Produzent: **AEGIS** DEVELOPMENT
2210 Wilshire Blvd., Nr. 27
Santa Monica, CA 90403
USA





Grafik von allen Seiten

Interessieren Sie sich für den IFF-Standard? Egal ob Sie eigene Programme schreiben wollen, die mit diesen Grafiken arbeiten, oder ob Sie Bilder direkt auf Diskette untersuchen wollen, unser Artikel über IFF (der Grafikstandard schlechthin) wird eine wertvolle Hilfe sein. Das Grafikformat ILBM wird anhand eines Demo-Bildes untersucht, wobei alle Bytes ausführlich angesprochen werden. Sollten Sie sich mehr für die Grafik-Hardware des Amiga interessieren, kommt unser neuer Kurs »Von Bit zum Bild« genau richtig. Darin werden zum Beispiel die Zusammenhänge zwischen Chips wie Copper und Blitter in bezug auf die am Bildschirm sichtbare Grafik dargestellt.

Ab Seite 57

Die Prominenz im Softwaretest

Diesmal sind einige bekannte Programme im Test, die für den Amiga umgesetzt wurden. Auf anderen Computern wie zum Beispiel dem C 64 ist Vizawrite immer noch ein Standard was Textverarbeitung betrifft. Kann »Vizawrite Desktop« für den Amiga diesen Erfolg wiederholen? »Datamat«, auch ein erfolgreiches C 64-Produkt, haben wir genauer untersucht. Mit dem Trend zu 3D-Grafik- und Animationssoftware sind zwei neue Programme auf den Markt gekommen: »Forms in Flight«, das sich mehr an die Einsteiger wendet und »Animator: Apprentice« für professionelle Animationen im Zeichentrickstil. Im Bild das neue Adventure »Arazok's Tomb«.

Ab Seite 118



AKTUELLES

EINSTEIGER

SPEICHERMEDIEN

KURSE

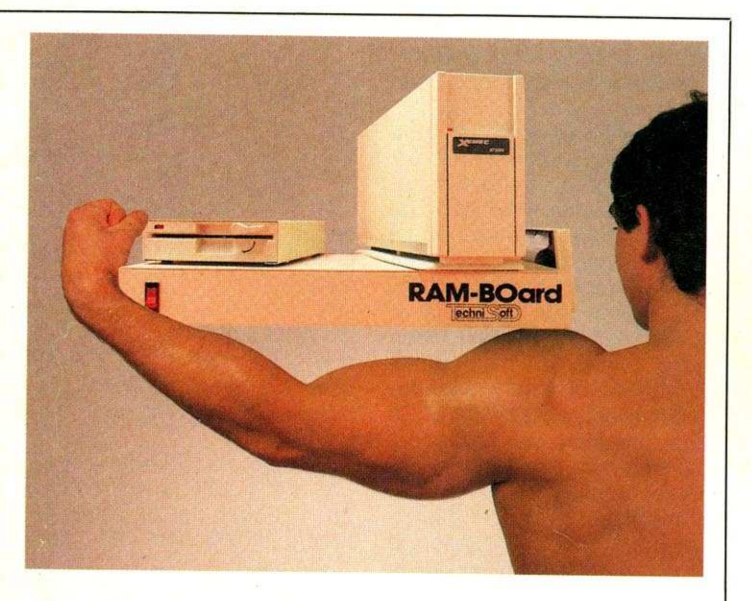
TIPS & TRICKS

SOFTWARE

HARDWARE

SOFTWARE-TEST

INHALT 12/87



Mehr Speicherplatz

Das Arbeiten mit dem Amiga wird leichter, wenn mehr RAM- und Massenspeicherplatz vorhanden sind. Ausführliche Grundlagen finden Sie zu Festplatten und Diskettenstruktur. Gerade eine mit Amiga-DOS formatierte Diskette weist einige interessante Besonderheiten auf. Eine Entscheidungshilfe beim Kauf bieten die Vergleichstests von Speichererweiterungen, Diskettenlaufwerken und Festplatten.
Ab Seite 22

AUFRUFE UND WETTBEWERBE

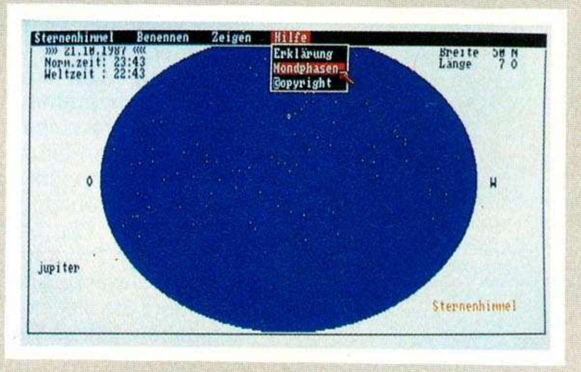
RUBRIKEN

■ Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind.

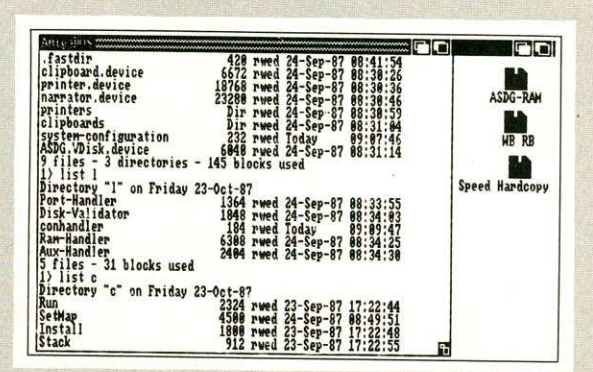
LISTINGS ZUM ABTIPPEN



Das Programm des Monats: »DCopy«, ein Kopierprogramm mit allem Drum und Dran



Himmelsbeobachtung von verschiedenen Orten wird zum Kinderspiel mit »Sternenhimmel«



Fantastisch schnelle Schwarzweiß-Ausdrucke in Superqualität macht »Speedhardcopy«



Tippfehler ade

Waren Sie nicht auch schon das eine oder andere Mal erstaunt, wenn nach mühevoller Abtipparbeit eines vielversprechenden Listings die einzige Meldung die des Fehlers war? Man vermutet zunächst eine Ungereimtheit im abgedruckten Programm, aber nach »nur« viermaligem Überprüfen des kompletten Listings hat man den Tippfehler auch schon entdeckt: ein O statt einer O. In Zukunft werden Sie vor derartigen Überraschungen bei den abgedruckten Listings im AMIGA-Magazin bewahrt bleiben. Wir haben für Sie »Checkie42« entwickelt. Sollten Ihre Finger anderer Meinung sein als der Ausdruck des Autors, sagt Ihnen unser »Checkie42« nicht erst am Schluß des Listings, sondern bereits am Ende der einzutippenden Zeile, ob Sie ein falsches Zeichen ausgewählt haben. Sie sind also immer darüber informiert, wenn nicht das richtige eingegeben wurde.

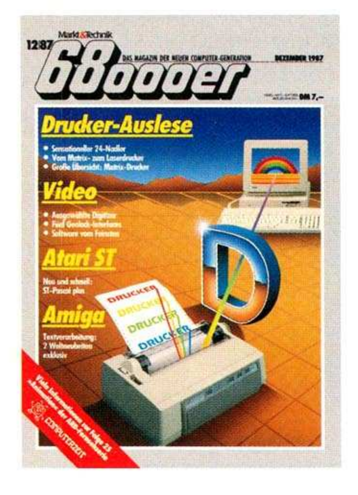
Und das Fantastische dabei ist: Checkie42 funktioniert bei allen Sprachen wie Basic, C, Modula, Pascal genauso zuverlässig wie bei Assembler. Das heißt, Sie verwenden in Zukunft für sämtliche Listings im AMIGA-Magazin nur einen Checksummer. Ein kleiner Schritt für die Redaktion (der erst einmal gemacht sein wollte), aber ein gewaltiger für die Leserschaft. Die Interpreter oder Compiler für die verschiedenen Sprachen sind natürlich nach wie vor notwendig. Die Zahl 42 stammt übrigens aus dem Buch The Hitchhiker's Guide to the Galaxy«. Insider wissen Bescheid? Outsider sollten vielleicht das Buch lesen.

In vielen Gesprächen und Zuschriften konnte ich immer wieder erfahren, daß sich manche Leser eine aktuellere Berichterstattung beziehungsweise »heißere« Tests im AMIGA-Magazin wünschen. Wir stehen auf dem Standpunkt, daß es wenig Nutzen bringt, die Beta-Version eines Produktes in einem ausführlichen Test vorzustellen. Zu unsicher sind die Aussagen, die sich zum endgültigen Verkaufsprodukt machen lassen. Was nützt es, sich über Abstürze und Unzulänglichkeiten in der Beta-Version auszulassen, wenn das Endprodukt ganz anders aussehen wird? Wie kann man über die Eignung einer Software für den deutschen Markt mit Umlauten und PAL-Adaption anhand der amerikanischen Vorabversion urteilen? Wie wird sich das ins Deutsche übersetzte Handbuch präsentieren? Es lassen sich zumeist lediglich Vermutungen anstellen. Vagheiten, halbseidene Statements oder Spekulationen wollen wir vermeiden.

Ausführliche Informationen zu neuen Entwicklungen, heißen Produkten, brodelnden Gerüchten oder interessante Interviews können Sie in Form von Messeberichten, Stories oder Kurzinformationen in unserem Aktuell-Teil finden. Wir besuchen für Sie die wichtigsten Messen im In- und Ausland. So waren beispielsweise die AmiExpo in New York City und die Systems '87 in München die herausragenden Ereignisse für den Amiga in diesem Herbst.

Wenn Sie der Meinung sind, daß wir halbfertige Produkte aus Gründen der Aktualität einem ausführlichen Test unterziehen sollten — auch das ist für Ihr AMIGA-Magazin kein Problem —, so schreiben Sie mir bitte. Natürlich auch dann, wenn Sie an fundierten, »antiquierten« Testberichten interessiert sind. Die Mehrheit bekommt wie immer Recht.

Albert Absmeier
Chefredakteur



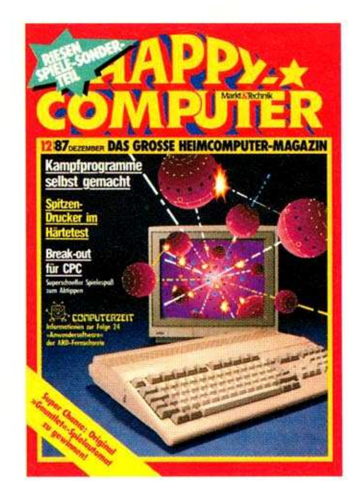
Die Systems '87 war in der Computerszene der Höhepunkt des Monats Oktober. Wir fanden tolle Neuheiten für den Amiga. Die Angebotspalette für die Hardware reicht vom Turbo-Amiga bis hin zum Echtzeit-Digitizer. Aber auch im Bereich der Software rührte sich einiges. Informieren Sie sich in der **Ausgabe 12/87** der 68000er über all diese aktuellen Neuheiten.

Bedingt durch seine aufwendige Hardware besitzt der Amiga unvorstellbare Fähigkeiten. Doch wie kann man als Programmierer von all diesen Fähigkeiten Gebrauch machen? Eine ausführliche Beschreibung der dazu notwendigen Hardwareregister bietet Ihnen unsere Referenzkarte.

Stehen Sie vor der Entscheidung, sich ein Textverarbeitungsprogramm für Ihren Amiga anzuschaffen? Unser Textverarbeitungs-Marathon zeigt Ihnen fünf Programme der unterschiedlichsten Preisklasse. Wo liegen die Stärken der Programme, und welche Schwächen haben sie? Nachdem Sie diesen Test gelesen haben, fällt es Ihnen leichter, sich für eine Textverarbeitung für Ihre Bedürfnisse zu entscheiden.

Die Qualität der Computeranimationen für den Amiga überzeugt immer mehr. Programme wie Video-Scape 3D oder Deluxe-Video machen Ihren Computer im Zusammenhang mit einem Genlock-Interface zum exzellenten Vorspanngenerator für Ihre Videofilme. Eine Übersicht über alle zur Zeit erhältlichen Genlock-Interfaces und Video-Digitizer, erläutert Ihnen die Einsatzbereiche und Funktionsweise.

Neben weiteren Tests und Berichten finden Sie eine Menge Informationen und Grundlagen zum Thema Computeranimation.



Die **Ausgabe 12/87** steht im Zeichen der PCW in London. Auf der wichtigsten englischen Messe gab es die heißesten Spiele und die neuesten Computer zu sehen. Amiga-Fans finden eine Umsetzung des bekannten Automaten »Missile Command« als Listing zum Abtippen. Welches Grafikprogramm ist das beste für den Amiga? Lohnt sich der Umstieg von Deluxe Paint II auf Prism oder Digi Paint? Ein ausführlicher Vergleichstest zeigt Stärken und Schwächen. Eine Herausforderung ganz neuer Art sind Kampfprogramme, die sich Schlachten im Speicher liefern. Durch die Amiga-Umsetzungen können Sie eigene Programme in die Arena schicken.

Die **Ausgabe 1/88** beweist, daß Computer Spaß machen. Mit Gag-Programmen zum Abtippen können Sie Freunde und Bekannte foppen. Wer möchte nicht lebensechte Bilder oder Trickfilme gestalten? Doch wie immer braucht man das richtige Programm dafür. Ein Vergleichstest zwischen Videoscape 3D, Deluxe Video 1.2 und Sculpt 3D zeigt, welches Programm für welchen Anwendungszweck am besten geeignet ist.

Ein Muß für alle Spiele-Fans ist der große Spiele-Sonderteil. Monat für Monat enthält er Tests der neuesten Spiele. Dieses Mal zum Beispiel das Autorennspiel »Test Drive« und die Umsetzungen »Western Games« und »BMX Simulator«. Wer Denkspiele liebt, wird auch das Listing »Keuz & Quer« für den Amiga mögen.

Was werden die Hits 1988? Zwölf Experten wagen den Blick in die Zukunft und enthüllen die Trends für 1988. Wie wird der Amiga abschneiden? Eines sei schon verraten: Seine Chancen 1988 sind trotz der starken Konkurrenz blendend.

Willkommen zur Amiga-Ausstellung

Eine eigene Messe hat dem Amiga noch gefehlt. In New York fand Anfang Oktober die erste AmiExpo statt. Hier konnte man sich konzentriert über das umfangreiche Angebot an Soft- und Hardware für den Amiga informieren.

Eine gute Idee und der feste Wille, diese auch durchzusetzen, war die eigentliche Geburtsstunde der ersten Amiga-Show in New York Anfang Oktober diesen Jahres. Nicht Commodore oder eine Messegesellschaft zeigten sich als Initiator, sondern ein ganz normaler Club, die »New York Amiga User Group«, kurz AMuse, mit Joseph Lowery an der Spitze, hat es geschafft, durch Engagement und Einsatz die meisten Anbieter und Hersteller für Amiga-Software und -Hardware im Sheraton Convention Center zu versammeln. Über 8000 Besucher informierten sich hier über das vielfältige Angebot rund um den Amiga.

Warum ein Amiga-Club eine eigene Messe veranstaltet? Die Mitglieder von AMuse sind überzeugt vom Amiga. Je bekannter der Amiga in allen Marktbereichen gemacht werden kann, egal, wie vertikal oder spezialisiert sie sind, desto mehr Software und Hardware wird für diesen Computer entwickelt. Mit mehr Entwicklern verbreitert sich auch die Anwendungsvielfalt für den »Endverbraucher«. Größere Märkte und viele Anbieter führen zu verstärktem Wettbewerb und damit letztendlich zu geringeren Preisen, was wiederum jedem Amiga-Besitzer zugute kommt.

Um diesen Mechanismus wie geschmiert weiterlaufen lassen zu können, sind die nächsten Amiga-Ausstellungen für 1988 ebenfalls schon terminiert: Los Angeles vom 16. bis 18. Januar; Chicago vom 22. bis 24. Juli und wieder New York City vom 8. bis 10. Oktober 1988. Jeder Amiga-Fan der zu diesen Zeiten in den Vereinigten Staaten verweilt, sollte mal einen Tag oder zwei auf den AmiExpos zubringen. Denn zu sehen und hören gibt es jede Menge.



Mitten im Herzen der Riesenstadt New York fand das Amiga-Ereignis statt: die erste AmiExpo, eine »Messe« nur für den Amiga

Bild 1. »The C 64 Emulator« von Readysoft für den Amiga. Mit einem zusätzlich zu erwerbenden Interface lassen sich auch die Peripherie-Geräte des C 64 anschließen.

Bunt und laut

Es kann an dieser Stelle nur ein kleiner Auszug aus dem vielfältigen Angebot der über siebzig Anbieter und Hersteller gegeben werden. Die Nennung von deutschen Bezugsquellen erscheint überflüssig, da die bundesrepublikanischen Distributoren und Händler sich sehr agil zeigen und nahezu jede neue Software mit nur wenigen Wochen oder Monaten Verzögerungen hierzulande anbieten. Da die Amerikaner kaum Exklusivrechte zugestehen, sind es immer auch mehrere Händler, die das Produkt zu unterschiedlichen Preisen anbieten.

Gerüchte um software- oder hardwaremäßige C 64-Emulatoren für den Amiga gibt es schon seit einiger Zeit. Auf der AmiExpo in New York konnten gleich zwei dieser Emulatoren besichtigt werden. Beide arbeiten ungefähr nach demselben Prinzip: Der C 64 wird rein durch Software emuliert. Zum Anschluß der C 64- und C 128-Laufwerke oder Drucker mit serieller C 64-Schnittstelle ist ein zusätzliches Interface-(Kabel) nötig. »The 64 Emulator« von Readysoft Inc. (Bild 1) kostet 39,95 Dollar ohne und 59,95 Dollar mit Interface. Das Programm ist nach Angaben des Herstellers 100prozentig in Maschinensprache geschrieben, unterstützt Sound, Farbe, Sprites und Rasterinterrupts

des C 64 sowie die Amiga-Laufwerke. Die 64er-Programme sollen mittels des Interface von der 1541/1581/1571 eingespielt und auf die Amiga-Laufwerke übertragen werden können. Vertreter des Unternehmens gaben an, daß es bei Schnelladesystemen und bei einigen Spielen sowie Business-Programmen zu Inkompatibilitäten kommen kann. Dennoch gaben sie die Anzahl der lauffähigen Programme mit 80 Prozent an. Näheres wird sicherlich ein ausführlicher Test des »The 64 Emulator« in einer der nächsten AMIGA-Ausgaben ergeben.

C 64-Emulatoren

Testen werden wir auch den zweiten Emulator, »GO 64!« von Software Insight Systems (Bild 2). GO 64! kostet mit Interface um die 100 Dollar. Auch hier soll wieder sämtliche C 64-Peripherie verwendet werden können.

Progressive Peripherals & Software kündigte mit »Access-64« ein serielles Interface für den Amiga 500 bis 2000 an. Damit sollen sich die Laufwerke 1541, 1571 und 1581 als DF8: oder DF9:, und die seriellen Drucker mit PRT: wie normale Amiga-Peripherie ansprechen lassen. Die mitgelieferte Software überträgt die Dateien der Text- oder Dateiverwaltung des C 64 auf den Amiga. Preis: 79,95 Dollar.

einem 68881-Coprozessor, getaktet von 14 bis 25 MHz, gesteckt und schon hat man eine 6000 bis 8000 Prozent höhere Leistung. Die Karte kostet zwischen 1495 und 1895 Dollar. CSA sieht einen so ausgerüsteten Amiga als preiswerte 3D Desktop-Workstation. Vergleichbare Workstations, wie etwa eine SUN 3/160, würden rund 60000 Dollar kosten. Die Berechnungszeit des in Bild 3 gezeigten Ray-Tracing-Bildes reduziere sich nach Auskunft eines CSA-Mitarbeiters von 63 Stunden bei einem normalen Amiga auf 40 Minuten. Auch für den Amiga 500 wird es eine »Piggypack« genannte Turbo-Karte geben. Sie soll ohne Coprozessor 595 Dollar und mit 68881-Matheprozessor ab 895 Dollar kosten.

Wer einen Amiga 500 hat, aber bereits Pläne hegt, auf den Amiga 2000 umzusteigen, dem hilft eine »Subsystem« genannte Erweiterung, sich die Karten für den 2000er zu kaufen und beim 500er zu verwenden. Das Subsystem (Bild 4) von Pacific Peripherals bietet zwei Amiga-Karten Platz, schmiegt sich genau unter den Amiga 500 und ist für 149 Dollar zu haben. Ein 3½-Zoll-Floppy-Laufwerk findet im Subsystem ebenfalls noch Aufnahme. Pacific Peripherals bietet auch noch eine SCSI-Interface für 249 Dollar an, mit dem man die externen Laufwerke des

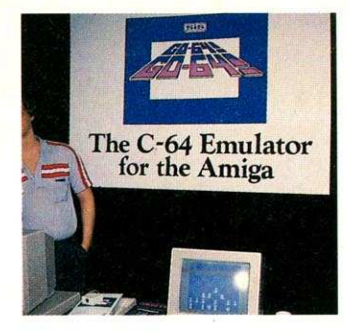


Bild 2. »GO 64!« ist ein C 64-Emulator für den Amiga.

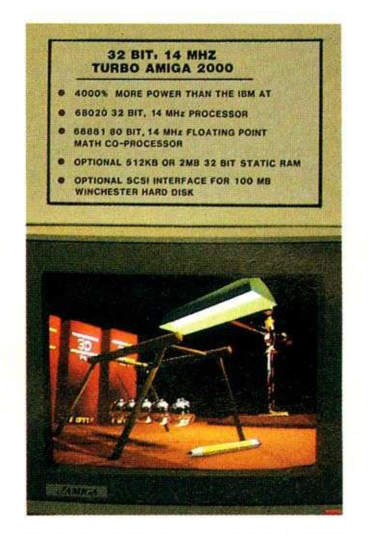


Bild 3. Die Turbo-Karte (mit 68020- und 68881-Prozessoren) wird einfach in den Amiga 2000 gesteckt. Bis zu 8000 Prozent soll der Amiga dann schneller sein.

Von Spirit Technology Corp. wurde für den Amiga 500 und 1000 eine interne Speichererweiterung auf bis zu 2,5 MByte mit Uhr und Kalender für 679 Dollar vorgestellt (Bild 6). Diese Erweiterungen kommen in den Prozessorsteckplatz. Beispielhaft ist die Dia-Show auf Diskette, die den genauen Einbau beschreibt.

Perry Kivolowitz, der Präsident von ASDG hat auf der AmiExpo einen sehr interessanten Satz geprägt: »The Amiga has moved from infancy to his childhood«. Frei übersetzt hat der Amiga sich gerade seiner Windeln entledigt und befindet sich auf dem Weg in seine Kindheit. Betrachtet man die Hardware, die bereits jetzt zum Amiga angeboten wird, fragt man sich, was kommen wird, wenn dieser Computer sich in die Pubertät stürzt.

Software in Hülle und Fülle

Allenthalben war auf der AmiExpo die explosionsartige Entwicklung der Angebotes an guter Software zu registrieren.

PageFlipper von Mindware erzeugt kleine Trickfilmsequenzen. Die Anzahl der IFF-Bilder ist nur durch den Speicher begrenzt. Im Lores-Modus werden 12 und im Hires-Modus sechs Bilder pro Sekunde abgespielt (Bild 7).



Bild 4. Auch der Amiga 500 kann jetzt zwei 2000er-Karten über das »Subsystem« (paßt direkt unter den Amiga 500) verwenden. Links zwei »ungewöhnliche« Laufwerke am Amiga.

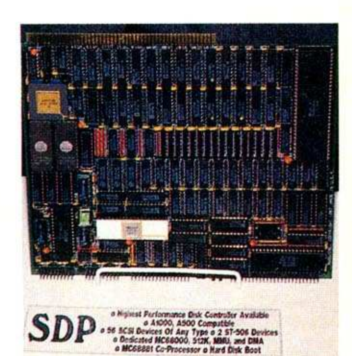


Bild 5. Bis zu 58 Laufwerke mit Übertragungsraten bis zu 400000 Byte pro Sekunde lassen sich über das SDP an den Amiga anschließen

Mit den Emulatoren stünde auf Anhieb eine Menge guter Software für den Amiga zur Verfügung. Wie viele der Programme tatsächlich ohne Probleme laufen, bleibt abzuwarten.

Die Beschleunigung der originalen Amiga-Software findet durch die Turbo-Karten von CSA statt. In den Amiga 2000 wird die 68020-CPU-Karte mit

Macintosh verwenden kann. Zudem ist die Wechselplatte von Infinity im Programm von Pacific. Die Grundkosten betragen 1295 Dollar, 10-MByte-Wechselplatten sind dann für je 18 Dollar zu haben.

Für die »high perfomance user« bietet die ASDG Inc. einen Satellite Disk Processor SDP (Bild 5) mit eigenem 68000-Prozessor und 512 KByte-Cache-Speicher für rund 1000 Dollar an. 56 SCSI-Laufwerke und zwei ST-506-Laufwerke lassen sich bei einer Übertragungsrate von bis zu 400000 Byte pro Sekunde an den A 500 und A 1000 anschließen. Gebootet wird von der Festplatte. Von ASDG gibt es auch noch Speichererweiterungen und Erweiterungsboxen für den Amiga 1000.

Aegis kündete eine PAL-Version mit deutscher Anleitung des Animations- und Grafikprogramms Videoscape 3D an (386 Mark). Besitzer einer originalen NTSC-Version können sich zwecks Umtausch zu üblichen Gebühren an ihren Händler oder an Softwareland A.G. in der Schweiz wenden.

Mit »Interchange« (49,95 Dollar) von Syndesis lassen sich Objekte sprich Bilder zwischen Sculpt 3-D und VideoScape 3D austauschen. Dadurch kann der einfacher zu bedienende Editor von Sculpt 3-D für VideoScape 3D-Bilder benutzt werden. Alle Interchange-Programme sind eigenständige Multitasking-Module. Syndesis hat weitere Module für zukünftige 3D-Programme angekündigt. So ist ein ständiger Austausch zwischen den unterschiedlichen 3D-Programmen gewährleistet.

Am Stand von Impulse herrschte stets großer Andrang, gab es hier doch ein neues Animations- beziehungsweise Ray-Tracing-Programm zu sehen. »Silver« zeichnet sich, so Stan Kalisher, der Ent-



Bild 7. Eine Szene aus einer mit PageFlipper erstellten Sequenz: Gleich gibt »er« »ihr« einen dicken Kuß

wicklungschef, durch besonders schnelle Berechnung der einzelnen Bilder aus. Es unterstützt alle Amiga-Grafikmodi, Genlock und PAL. Der Preis liegt bei 169,95 Dollar.

»Forms In Flight« von Micro Magic gibt es in einer verbesserten Version Rev 1.1. Das 3D-Zeichen- und Animationsprogramm erfordert mindestens 1 MByte und kostet 79 Dollar.

NewTek, der Hersteller von DigiPaint und DigiView, ließ am Stand die hübsche Dame in Bild 8 sich in digitalisierter Form bewegen und sprechen. Dies sollte ein erster Hinweis für sich abzeichnende neuartige Videoprogramme von NewTek sein.

Echtzeitdigitalisierer waren ein Schlagwort auf der Messe. »Perfect Vision« von SunRize Industries und »Live!« von A-Squared waren die herausragenden Beispiele dafür. Perfect Vision kostet 219,95 Dollar, stellt in 16 Graustufen oder 4096 Farben dar. Die Digitalisierer oder Frame Grabber sind derzeit nur in der NTSC-Version erhältlich, aber die Richtung ist vorgegeben.

Lattice oder Manx

Es kommt fast einem Ideologienstreit gleich, wenn es um die Frage geht, für welchen C-Compiler man sich entscheiden soll. Die zwei Hauptkonkurrenten stellten auf der AmiExpo neue Versionen der bewährten »Schlachtrösser« vor. Lattice gab die Version 4.0 Amiga-DOS-C-Compiler frei. Die neueste Version erzeugt durch das Inline-Interface zum Amiga-ROM kleinere und effektiver ausführbare Programme. Der Compiler arbeitet mit 16- oder 32-Bit-Integer. Die Bibliothek enthält nun mehr als 250 Amiga-, ANSI-, Unix und Lattice-Funktionen. Der Auf-

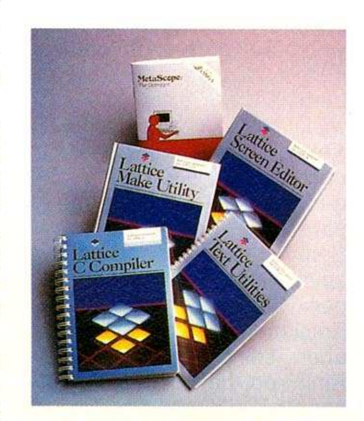


Bild 9. Das komplette Lattice-Amiga-DOS-C-Entwicklungssystem in der Compiler-Version 4.0

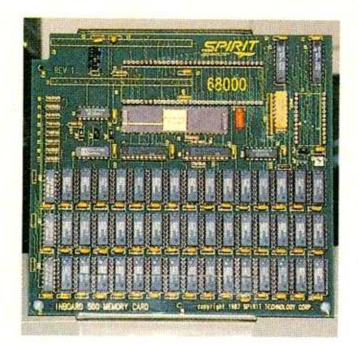


Bild 6. Interne Speichererweiterungen auf bis zu 2,5 MByte für den Amiga 500 und 1000 von Spirit

rüstpreis beträgt 75 Dollar, der neue Compiler allein kostet 200 Dollar, das Entwicklungspaket 375 Dollar (Bild 9).

Manx Software Systems kontert mit der Version 3.6 des Aztec C68k für den Amiga. Neu dabei ist der Debugger auf Quellenebene, der Manx Aztec SDB. Daß sich dadurch die Entwicklungszeit erheblich reduziert und dem Anfänger der Einstieg in die C-Programmierung erleichtert wird, liegt auf der Hand. Der Source Level Debugger schlägt mit 75 Dollar zu Buche. Das Aztec C68k/Am 3.6-Entwicklungssystem mit Unix-Utilities und Bibliotheken für den 68881 kostet 299 Dollar. Welcher Compiler für wen am besten geeignet ist, wird ein ausführlicher Test in einer der nächsten Ausgaben ergeben.



Bild 8. Digitalisierte Sprache und dazu passende Mundbewegung demonstrierte diese Schönheit bei NewTek

Auf dem Gebiet des Desktop Publishing ist der Amiga gewaltig im Kommen. Die bisher erschienenen Programme dürfen getrost als Vorreiter einer neuen verbesserten Generation angesehen werden. »Professional Page« (Bild 10) von Gold Disk erlaubt es dem Anwender sogar, Farbgrafiken in die fertige Seite einzuarbeiten. Eine vollwertige WYSIWYG-Textverarbeitung, Ausgabe auf Postscript-Geräten, CAD-Grafiken mit eigenen Befehlen, 999 Seiten pro Dokument und professionelle Seitenbearbeitungsfähigkeiten sind nur einige Features von Professional Page. Erhältlich soll es noch vor Weihnachten zum Preis von 395 Dollar sein.

»Shakespeare« von Infinitiy ist das neueste Mitglied der Desktop Publishing-Programmgemeinde. Besonders bemerkenswert ist die durchgehend farbige Gestaltung der Grafiken (aus Deluxe Paint) und des Textes. Anfang Dezember soll es mit deutschem Handbuch und Umlauten bei uns zu kaufen sein. Der Preis in den USA beträgt 225 Dollar.

Spiele

Auf dem Gebiet der Lernsoftware tut sich einiges. So demonstrierte beispielsweise P.S. Squared Ltd. das Programm »Math-Animation« (Bild 11). Hätte ich dieses farbige Programm während meines Studiums gehabt, die Mathe würde sicherlich noch wesentlich mehr Spaß gemacht haben. In Math-Animation stecken vier Mannjahre an Entwicklungszeit. Im Februar nächsten Jahres wird es endgültig fertiggestellt und auch die Preisfindung beendet sein. Gespannt darf man auf alle Fälle sein.

Die Spiele waren erstaunlicherweise auf der AmiExpo erheblich unterrepräsentiert. Selbstverständlich waren Firmen wie Activision, Infocom, Firebird, Psygnosis, Electronic



Bild 10. Professional Page gilt als das derzeit beste Desktop Publishing-Programm auf dem Amiga. Farbige Grafiken im Text sind sicher nicht alltäglich.

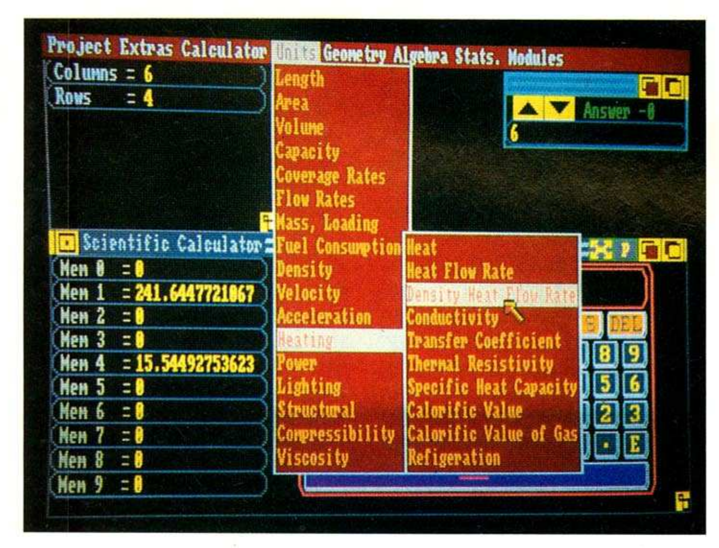


Bild 11. Mathematik kann auch farbenfroh sein und Spaß machen, wie »Math-Animation« von P.S. Squared beweist. Das Programm wird im Februar '88 auf den Markt kommen und für jeden Schüler und Studenten eine wertvolle Hilfe darstellen.

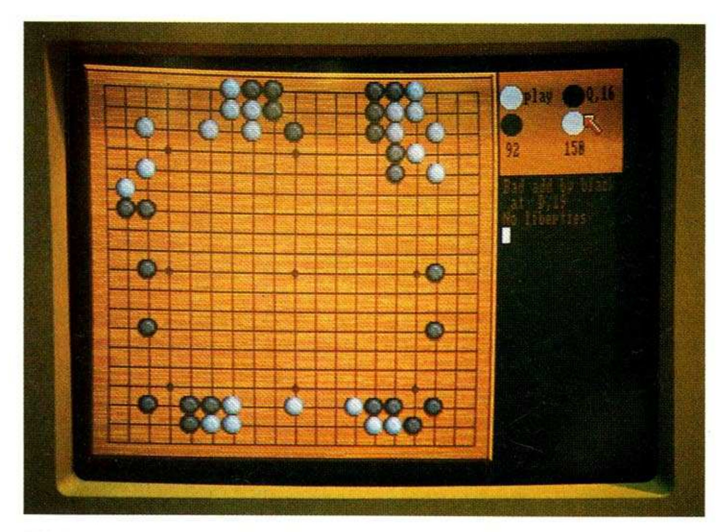


Bild 12. Das asiatische Brettspiel »Go« gibt es jetzt von Infinity für den Amiga. Im 18. kyu-Level ist Go selbst von Experten nur noch schwer zu schlagen.

Arts und Discovery präsent, aber sie gingen in den »ernsthaften« Anwendungen fast unter. Dennoch gab es das eine oder andere neue Spiel zu bestaunen, wie beispielsweise Go von Infinity (Bild 12) oder Arkanoid von Discovery Software. Man will sich bei Discovery in Zukunft mehr in Richtung Spiele bewegen; nur zur Erinnerung, Marauder II ist aus dem Hause Discovery. Sublogic zeigte die Scenery Disk 7 und 11 zum Flight-Simulator.

Eine neue Art von Spielen wurde mit TeleGames vorgestellt. Wie der Name vermuten läßt, wird hier via Modem oder Akustikkoppler gespielt. Allerdings agiert auf der anderen Seite nicht wie sonst üblich ein Computer, sondern in diesem Fall ein Mensch. Schach, Dame, Backgammon und ein Taktikspiel auf dem Schlachtfeld sind derzeit für 34,95 Dollar erhältlich. Software Terminal, der Hersteller, will nach eigenen Angaben mit diesen Produkten nach Deutschland kommen, sobald es hier genügend legal betriebene Modems gibt.

Selbstverständlich spielte die Musik eine wichtige Rolle bei den vorgestellten Programmen. Einige seien hier kurz erwähnt: Studio Magic von SunRize, Texture Version 2.5 von Sound Quest, Perfect Sound von MicroSearch, SoundScape von Mimietics, Master Librarian V 2.00 mit Editoren für Roland und Yahmaha-Synthesizer von Sound Quest, Deluxe Musik von Electronic Arts, und Musik Mouse von Opcode Systems. Für nähere Informationen sind die Kontaktadressen angeführt.

Es hat sich gelohnt

Die AmiExpo ist ein voller Erfolg geworden. Über 8000 Besucher in drei Tagen haben die vielen Seiten des Amiga gesehen. Aus allen Anwendungsbereichen gab es neue Entwicklungen oder zumindest Ankündigungen für das Weihnachtsgeschäft. Neben der eigentlichen Ausstellung gab es eine Reihe von Workshops und Seminaren. So erzählte beispielsweise Jay Miner (Bild 13), der »Vater« des Amiga über die Entstehung des Amiga und über seine Zukunft. Diese AmiExpo hat das oben genannte Ziel der New York Amiga User Group voll erfüllt. Nun ist der Amiga nicht mehr aufzuhalten, und die nächste AmiExpo kommt bestimmt. (aa)

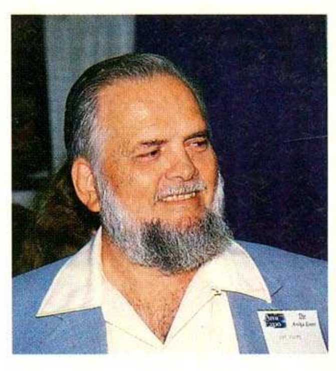


Bild 13. Jay Miner gilt als der »Vater« des Amiga. Er hielt einen interessanten Vortrag über die Genesis des Amiga.

Bezugsquellen:

Readysoft Inc., P.O. Box 1222, Lewiston, N.Y. 14092

Software Insight Systems, 122 Prospect Hill Road, East Windsor, CT 06088

Progressive Peripherals & Software, Inc., 464 Kalamath Street, Denver, Colorado 80204

CSA, 7564 Trade Street, San Diego, CA 92121

Pacific Peripherals, P.O. Box 14575, Fremont, CA 94539

ASDG INC., 280 River Road Suite 54-A, Piscataway, NJ 08854

Spirit Technology Corporation, 220 West 2950 South, Salt Lake City, Utah 84115

Mindware International, 110 Dunnlop St. W., Box 22158, Barrie, Ontario, Canada L4M 5R3

Aegis Development Inc., 2210 Wilshire Blvd., Suite #277, Santa Monica, CA, 90403

Syndesis, 20 West Street, Wilmington, Massachusetts

Impulse, Inc., 6870 Shingle Creek Parkway, Suite 112, Minneapolis, MN. 55430

Micro Magic, Suite 320B, 261 Hamilton Av., Palo Alto, CA 94301

NewTek, Inc., 701 Jackson, Suite B3, Topeka, KS 66603

Lattice, Inc., 2500 South Highland Avenue, Lombard, IL 60148

Manx Software Systems, Box 55, Shrewsbury, NJ 07701

Gold Disk, P.O. Box 789, Streetsville, Mississauga, Ontario, Canada L5M 2C2

Infinity Software, Inc., 1144-65th Street, Studio C, Emeryville, CA 94608

P.S. Squared Ltd., "Zion", Rossacon, Newmarket, Co. Cork, Ireland

Discovery Software International, Inc., 163 Conduit Street, Annapolis, MD 21401

Opcope Systems, 1024 Hamilton Ct., Menlo Park, CA 94025

SunRize Industries, 3801 Old College Road, Byran, Texas 77801

Magnetic Music, Rd. 5 Box 227A Myrtle Dr. Mahopac, N.Y. 10541

Sound Quest Inc., 5 Glenaden Avenue East, Toronto, Canada M8Y 2L2

Mimietics Corporation, P.O. Box 60238, STA. A. Palo Alto, CA 94306

MicroSearch, 9896 Southwest Freeway, Houston, Texas 77074

Bewegung im Druckermarkt

Viele Hersteller stellten auf der Systems dem Publikum ihre neuesten Drucker vor. Zahlreiche Weltmeister waren zu bestaunen. Die wichtigste Entwicklung: Die Leistung steigt, die Preise sinken.

Der entscheidende Trend bei den Matrixdruckern: Die 24-Nadel-Drucker sind auch im unteren Preissegment im Kommen. Sowohl Epson als auch NEC präsentierten einen Drucker mit 24-Nadel-Druckkopf für deutlich unter 1500 Mark.

NEC stellte den Pinwriter P2200 (Bild 1) sogar schon kurz vor der Systems in München der Presse vor. Einige hervorstechende Merkmale dieses Druckers sind:
— der eingebaute Schub- und Zugtraktor,
— die Papierparkfunktion zur gleichzeitigen Verwendung von Einzel- und Endlospapier,
— sechs verschiedene Schriftarten,
— vielfältige Programmierfunktionen und
— die hohe grafische Auflösung von bis zu 360 x 360 Punkten pro Zoll.

Der Drucker ist kompatibel zu den Pinwriter-Modellen P5, P6 und P7. Er verfügt standardmäßig über eine Centronics-Schnittstelle und ist einfach an den Amiga anzuschließen. Der Preis des Pinwriter 2200 beträgt zur Zeit 1138 Mark inklusive Mehrwertsteuer.

Epson ergänzt seine Produktpalette bei den 24-Nadeldruckern um die im September vorgestellten LQ-850 und LQ-1050, sowie den neuen LQ-500 (Bild 2).

Die Drucker LQ-850 und LQ-1050 verfügen über ein Papierhandling, das es ebenfalls erlaubt, Endlos- und Einzelblattpapier parallel zu bearbeiten. Beide Drucker verfügen über einen eingebauten Schubtraktor, zwei Schönschriftmodi und komfortable Bedienungselemente zur Druckerprogrammierung.

Mit dem LQ-500 will Epson die 24-Nadel-Qualität auch für den semiprofessionellen Bereich zugänglich machen. Aus Kostengründen verfügt dieser Drucker jedoch nicht über das praktische Papiermanagement der beiden anderen Epson-Modelle, sondern über einen Zugtraktor. Die beiden NLQ-Schriftarten »Roman« und »Sans Serif« kann der LQ-500 zusätzlich in den Variationen »Outline« und »Shadow« darstellen. Alle drei Drucker verfügen über eine Centronics-Schnittstelle und unterstützen den weitverbreiteten Epson ESC/P-Standard.



Bild 1. Der Pinwriter P2200 von NEC druckt mit 24 Nadeln.

Epson gibt zwar auch den 9-Nadel-Druckern noch eine Überlebenschance, jedoch nur im unteren Preissektor. Es ist anzunehmen, daß durch die Einführung des LQ-500 auch die Preise für die preiswerteren 9-Nadel-Drucker, zum Beispiel des Epson LX- 800, noch weiter sinken.

Der genaue Preis des Epson LQ-500 stand zu Redaktionsschluß noch nicht fest. Der Drucker wird höchstwahrscheinlich im Dezember für etwa 1100 Mark in den Handel kommen.

Auch Fujitsu setzt auf 24 Nadeln. Der laut Angaben des Herstellers weltschnellste 24-Nadel-Drucker, der Fujitsu DL5600, erreicht Druckgeschwindigkeiten von bis zu 486 Zeichen je Sekunde. Diese hohe Leistung hat jedoch ihren Preis: 6099 Mark kostet der DL5600.

Preisgünstiger und besonders für den Amiga interessant sind die Drucker DL3300 und DL3400. Beide können mit einem Umrüstsatz farbig drucken. Der Preis ohne Farbkit beträgt 2550 beziehungsweise 2650 Mark. Wie wir erfahren konnten, wird an eigenen Druckertreibern für den Amiga gearbeitet.

Bild 2. So präsentierte Epson den LQ-500 auf der Messe

Aufsehen erregte die Präsentation des ersten Star-Laserdruckers. Der Laserprinter 8 soll besonders durch seine niedrigen Betriebskosten (angegeben sind 7,5 Pfennig je Blatt) Marktanteile erobern. Erste Lieferungen des Druckers sind für Januar 1988 geplant. Der Endverkaufspreis dürfte bei 6500 Mark liegen. Ansonsten setzt Star auf seine bekannte Druckerpalette.

Seikosha stellte nicht nur den Motorrad-Weltmeister Toni Mang, sondern auch den angeblich schnellsten Matrixdrucker der Welt vor. Der SBP-10 soll mit einem 18-Nadeldruckkopf eine Geschwindigkeit von 800 Zeichen pro Sekunde erreichen.

Für Dezember kündigte Seikosha die Auslieferung des 24-Nadel-Druckers SL-130 an. Einige Merkmale dieses Druckers:
— Schreibbreite 15 Inch
— Papierparkfunktion
— Einzelblattschacht
— Schubtraktor

Der SL-130 soll 1899 Mark kosten.

Seikosha beabsichtigt, auf der CeBIT 1988 einen neuen Seitendrucker zu präsentieren. Er wird, wie zu hören war, über drei Emulationen verfügen: LaserJet, Epson und IBM.

Laut Ankündigung wird dieser Drucker etwa 5000 Mark kosten.

Auch zu diesem Preis bleibt die Anwendung der Laserdrucker auf professionelle Bereiche, beispielsweise Desktop Publishing und CAD, beschränkt.

Das größte Wachstumspotential im Bereich um 1000 Mark und darüber haben zur Zeit die 24-Nadel-Drucker. Die 9-Nadel-Drucker geben jedoch im unteren Preisbereich noch eine Zeitlang den Ton an.

Die neue Druckergeneration druckt schneller, druckt in höherer Auflösung und preiswerter. Der Bedienungskomfort steigt: Papierparkfunktionen, Schub- und Zugtraktoren, viele Schriftarten und mannigfaltige Funktionen zur Druckerprogrammierung gehören schon zum Standard. Und Farbe? Die Hersteller konzentrieren sich noch auf den Bereich »Schwarzweiß«. Es bleibt zu hoffen, daß gerade für den Amiga auch bald 24-Nadel-Drucker in Farbe für unter 1500 Mark angeboten werden. (ub)

Computer des Jahres: Amiga 500

Zur Pressekonferenz von Commodore auf der Systems '87 konnte der Amiga 500 als Computer des Jahres 1987 präsentiert werden (Bild 3). Alle Commodore-Produktlinien wurden von den bekannten Spielern des FC Bayern München enthüllt. Mit berechtigtem Stolz konnte Commodore-Geschäftsführer Hoffmann außerdem die Auslieferung des 100 000sten Amiga in Deutschland bekanntgeben. Allein vom Modell Amiga 2000 wurden im ersten Quartal fast 7000 Einheiten ausgeliefert. Auch das Profigerät aus der Amiga-Reihe scheint sich langsam aber unaufhaltsam am Markt zu etablieren. (jk)



Bild 3. Winfried Hoffmann gab die Auslieferung von 100 000 Amigas bekannt

Raytracing mit Animation

»Silver« heißt das neue Raytracing-Programm von Impulse, das auch auf der Systems vorgestellt wurde (Bild 4). Zum Programm gehört ein komfortabler Szeneneditor, mit dem sich die berechneten Bilder nacheinander durchschalten lassen, um so eine Animation zu erhalten. Silver kann maximal 512 solcher Bilder für eine Szene verarbeiten. Die Schnelligkeit, mit der die einzelnen Grafiken angezeigt werden, kann von 6 bis 60 Bilder pro Sekunde variiert werden. Silver arbeitet in allen Auflösungsstufen des Amiga und unterstützt auch den Overscan-Modus. Einer Zusammenarbeit des Programms mit einem Genlock-Interface steht nichts im Wege, trotzdem wurde zusätzlich ein Nachladen von Hintergrundgrafiken im IFF-Standard vorgesehen. Die Programmierer von Silver rühmen sich, daß selbst komplett durchgerechnete Raytracing-Bilder in nur vier Minuten vollständig sichtbar sind. Dem werden wir in einer der nächsten Ausgaben der AMIGA nachgehen. Silver soll für umgerechnet 348 Mark in Deutschland angeboten werden. (jk)

Intelligent Memory, Basaltstraße 58, 6000 Frankfurt 90, Tel. 069/7 07 11 02

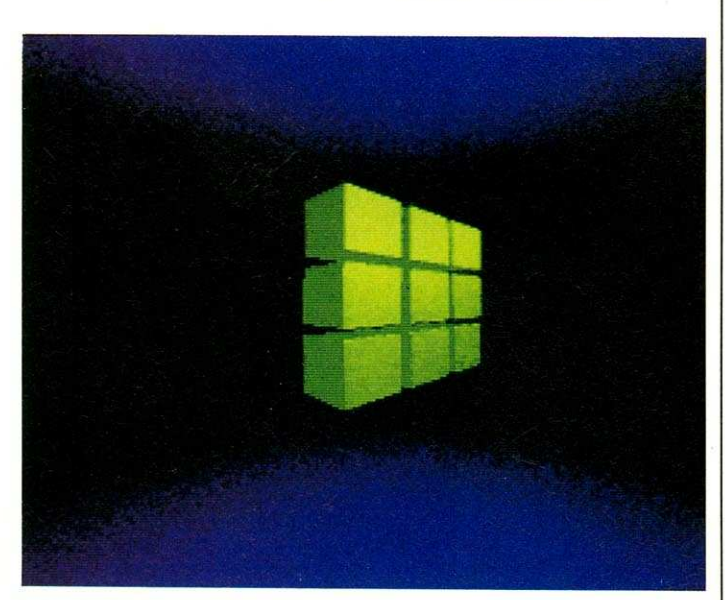


Bild 4. »Silver« verheißt schnelles Raytracing

Video-Digitizer

Neu ist der von Merkens angebotene S/W- und Farbvideo-Digitizer VD-3 AMIGA. Die Erfassung eines Bildes erfolgt in 20 Millisekunden. Das digitalisierte Bild wird über eine Parallelschnittstelle an den Amiga übertragen. Es lassen sich, je nach Datenmenge, bis zu zehn Bilder pro Sekunde übertragen.

Da der Digitizer auch den Interlace-Modus des Amiga unterstützt, ist eine maximale Auflösung von 640 x 512 Punkten möglich. Die Software, die zum VD-3 AMIGA geliefert wird, erlaubt die Darstellung der digitalisierten Bilder in 16 Graustufen oder 32 Echtfarben. Im HAM-Modus werden 4096 Farben wiedergegeben.

Kontrast und Helligkeit der Bilder sind softwaresteuerbar. Der Video-Digitizer verfügt über vier schaltbare Video-Eingänge und einen Triggereingang. Der Preis des VD-3 AMIGA inklusive Software und der für die Farbbild-Digitalisierung erforderlichen Farbscheiben beträgt 1798 Mark. Ein zusätzliches schaltbares RGB-Farbfilter, das ein RGB-Signal in die drei Grundfarben splittet, so daß keine Farbscheiben mehr nötig sind, kostet 648 Mark. Ein vollautomatischer RGB-Splitter, der ein müheloses Digitalisieren von Farbbildern erlaubt, ist voraussichtlich ab November lieferbar. (ub)

Merkens, Elektronische Datenverarbeitung, Fuchstanzstraße 6A, 6231 Schwalbach, Tel. (061 96) 30 26

DTM-News

Einige interessante Programme für den Amiga wurden von DTM und Intelligent Memory, die als Unteraussteller am Commodore Stand vertreten waren, vorgestellt. Als erstes wäre der Diskettenmonitor »Discovery« von Ralph Babel zu nennen, der jetzt in einer fertigen Version vorliegt. Dabei stehen sieben verschiedene Module zum Untersuchen der Files, Tracks und Sektoren einer Diskette zur Verfügung. Spezielle Funktionen zum Untersuchen der Diskette auf eventuelle Fehler runden das Programm ab. Der voraussichtliche Verkaufspreis wird bei 198 Mark liegen.

Weiterhin wurde das Mathematikprogramm »Funktion« angekündigt, das bis zu 50 Funktionen gleichzeitig in linearen, exponentiellen oder logarithmischen Koordinatensystemen darstellen soll. Automatische Berechnung der Ableitungen sollen auch integriert sein. Der angekündigte Verkaufspreis ist 98 Mark.

Der »Page Flipper« (siehe auch Bericht AmiExpo) wird in Deutschland von DTM zum Preis von 98 Mark verkauft.

Zum Animator: Apprentice gibt es inzwischen Zusatzdisketten mit verschiedenen Objekten und Schriften sowie den »Animator: Flipper«, mit dem sich die einzelnen Grafiken animieren lassen. Der Preis je Zusatzdiskette beläuft sich auf 69 Mark, während der Flipper 129 Mark kosten soll.

Als letztes wären noch drei Spiele zu erwähnen: Testdrive ist ein Autorennspiel zum Preis von 89 Mark. Bei dieser Fahrsimulation, die von Accolade entwickelt und über Electronic Arts vertrieben wird, können verschiedene Superautos auf gefährlichen Kursen gesteuert werden. Auswählen darf man beispielsweise zwischen Marken wie Porsche, Lotus, Lamborghini und Corvette. Sogar Gangschaltung, vorbeifahrende Autos im Rückspiegel und Fliegendreck auf der Windschutzscheibe sind integriert. »Firepower« bietet dreidimensionale Action mit Panzern für 59 Mark und Detonator ist eine preiswerte »Break Out«-Variante mit vielen Extras und gutem Sound für 49,95 Mark. (jk)

DTM, Poststraße 25, 6200 Wiesbaden, Tel. 06121/560084
Intelligent Memory, Basaltstraße 58, 6000 Frankfurt 90, Tel. 069/7071102

Marvin-Scanner

Print-Technik bietet seit kurzem einen Flachbett-Scanner auch für den Amiga an (Bild 5). Der Scanner kann auf dreierlei Arten genutzt werden:
1. Zur Bilderfassung
2. Als Kopierer
3. Als Thermo-Drucker

Im Druckermodus können nach Angaben des Herstellers Texte mit einer Geschwindigkeit von 40 Zeichen pro Sekunde ausgegeben werden. Die Druckdichte beträgt 200 Punkte je Zoll.



Der Marvin-Scanner kann mehr als nur Bilder erfassen

Im Scannerbetrieb werden DIN-A4-Vorlagen mit einer Auflösung von ebenfalls 200 Punkten pro Zoll erfaßt. Das Einlesen der Vorlage dauert etwa zehn Sekunden. Die Darstellung erfolgt in 16 Graustufen. Sie können die Bilddaten im IFF-Format speichern. Somit können Sie die gescannte Vorlage in alle gängigen Mal-, Zeichen- und Desktop Publishing-Programme übernehmen. Das Grundmodell kostet 2998 Mark. (ub)

Print-Technik, Nikolaistr. 2, 8000 München 40

Kommerzielle Anwendungen nicht nur für Amiga-PC

Ein in kaufmännischen Bereichen eingesetzter Amiga hatte bisher in der Regel die Modellbezeichnung 2000 und verfügte über eine PC-Karte, mit der MS-DOS-Software betrieben werden kann. So läßt sich einerseits das große Angebot an Programmen für den IBM-Standard und andererseits die überlegene Hardware des Amiga für Spezialanwendungen nutzen. Gerade die grafische Leistungsfähigkeit dieser Hardware macht eine schnelle und komfortable Benutzeroberfläche möglich. Der Anwender eines Programms benötigt weniger Computerwissen und kann sich bei der Bedienung auf das Wesentliche konzentrieren. Die Tatsache, daß der Amiga diese Benutzeroberfläche bereits im Betriebssystem integriert hat, scheint den Softwareentwicklern Grund genug zu sein, nun unabhängig von der PC-Seite hochkarätige Programme für den Amiga zu entwickeln oder zu übertragen.

Einige dieser Softwarehersteller stellten Ihre Produkte auf der Systems vor. Mit einer der ersten Anwendungen im kommerziellen Bereich — der Textverarbeitung — war die WordPerfect Software GmbH vertreten. Das für 790 Mark vertriebene »WordPerfect« ist die erste Textverarbeitung auf dem Amiga, die sich mit dem am PC gesetzten Standards wie Word oder Wordstar messen kann. Hervorstechende Merkmale dieses Programms sind Fußnotenverwaltung, mehrspaltige Darstellung, der Einsatz von Makrobefehlen sowie »Thesaurus« — ein Hilfsprogramm zur Darstellung von Synonymen und Antonymen. In der nächsten Ausgabe finden Sie einen ausführlichen Test von WordPerfect.

Mit »AmigaBuch« stellten Oliver Fendt (Programmierer) und Dr. Ralf Schwarz (Steuerberater) eine leistungsfähige Finanzbuchhaltung vor. Das Programm benötigt einen auf 1 MByte Speicher aufgerüsteten Amiga und ist in drei Versionen erhältlich. Alle drei Versionen enthalten Module für die Sachkonten- und Lieferantenverwaltung, zum Buchen und die Ausgabe der Auswertungen. Gegenüber der Version 1 (1000 Mark) enthalten die Versionen 2 (1900 Mark) und 3 (2750 Mark) zusätzliche Module für die Kunden- und Artikelverwaltung sowie Lieferschein- und Auftragskontrolle. Je nach Version lassen sich damit 50/32000 Kunden beziehungsweise 400/32000 Artikel verwalten. AmigaBuch verzichtet weitgehend auf Nummern — die einzelnen Konten lassen sich über die Namen der Lieferanten, Kunden oder Artikel aufrufen. Die Lagerverwaltung soll sich mit der Bildung sogenannter Sets (Beispiel: Computer und Monitor) und der korrekten Abbuchung der einzelnen Bestandteile vereinfachen. Ein hauptsächlich für Steuerberater entwickeltes Umbuchungsprogramm bietet Funktionen, welche die Auswirkung manueller Umbuchungen auf den Gewinn anzeigen oder aber Hauptübersicht und Umbuchungsjournal drucken.



Bild 6. »Amiga Calc«, die neue Tabellen-Kalkulation

Ebenfalls in diese Marktnische zielt »Bureau Amiga« der Edotronik GmbH aus München. Das preisgünstigere Paket (FiBu, Lohn und Auftragswesen jeweils 700 Mark; Text-Adreva 300 Mark) dürfte in den Feinheiten nicht so ausgeklügelt sein wie AmigaBuch, bietet aber ein komplettes System für die kaufmännische Verwaltung eines Kleinbetriebes.

Ein Tabellenkalkulationsprogramm (Bild 6) präsentierte DOS-Software aus Berlin. Bei der Vorführung von »AmigaCalc« beeindruckte die Integration der Amiga-Benutzeroberfläche. Durch den intensiven Einsatz von Auswahlgadgets haben die Entwickler ein einfach zu bedienendes Programm erstellt. Für Fachleute, die mit dem monochromen Mikrocomputer aufgewachsen sind, wirkte es dadurch an manchen Stellen zwar etwas farbig, aber gerade diese optische Gestaltung erleichtert die Orientierung. Durch den mehrdimensionalen Aufbau sollen sich Daten zwischen den einzelnen Kalkulationsblättern austauschen lassen. AmigaCalc ist in Berlin entwickelt worden und enthält somit eine, wenn auch nur knapp gehaltene, deutsche Dokumentation. Zum Redaktionsschluß war nur der voraussichtliche Preis von 248 Mark bekannt. (pa)

WordPerfect: WordPerfect Software GmbH, Frankfurter Str. 33-35, 6236 Eschborn, Tel. 06196/481722
AmigaBuch: Apraxas GmbH, Rottenbucher Straße 2, 8032 Gräfelfing
Bureau Amiga: Edotronik GmbH & Co KG, St.-Veit-Straße 70, 8000 München 80, Tel. 089/404093
AmigaCalc: DOS-Software, Carmerstraße 7, 1000 Berlin 12, Tel. 030/3177224

Video-Genlock

Merkens präsentierte das IVS-Genlock 1000 für den Amiga 500, 1000 und 2000. Dieses laut Hersteller vollprofessionelle Video-Genlock bietet großen Bedienungskomfort. Beispiele sind unter anderem:
— Farbauswahlschalter ermöglichen es, nur bestimmte Farben des Amigabildes sichtbar zu machen.
— Ein- und Ausblendungen lassen sich stufenlos regeln.
— Das Hintergrundvideosignal läßt sich auf weiß oder schwarz ausblenden.

Das IVS-Genlock 1000 verfügt über zwei Videoeingänge, einen Computervideoausgang (reines RGB-Signal des Amiga) und einen Mischvideoausgang. Mit dem neuen »Digital Locking«-Verfahren, setzt das Genlock die Amiga-Grafik fest ins Bild. Der Preis: 4548 Mark. (ub)

Der Turbo ist da!

Allen Amiga-Besitzern, die mehr Rechenleistung und -Kapazität benötigen, ist mit dem neuen Turbo-Amiga geholfen (Bild 7). Die A2000-Zusatzkarte, die in den 86-Pin-Slot (dieser enthält normalerweise die 512-KByte-RAM-Erweiterung) gesteckt wird, beschleunigt die normale Arbeitsgeschwindigkeit des Amiga-Systems um 200 bis zu 700 Prozent. Dies gilt allerdings nur wenn Speichererweiterungen mit 32 Bit Busbreite und 14 MHz Taktfrequenz eingebaut sind. Die Karte ist mit einem 68020-Prozessor bestückt, der mit einem Takt von 14 MHz arbeitet. Zusätzlich ist ein Arithmetik-Coprozessor vom Typ 68881 enthalten, der entweder mit 14 MHz (Beschleunigung: 1100 Prozent; Preis der Karte: 3100 Mark) oder, in einer »größeren« Ausführung der Karte, mit 20 MHz (Beschleunigung: 1450 Prozent; Preis der Karte: 4995 Mark) Taktfrequenz läuft. Ermittelt wurden diese Werte mit einem Public Domain-Mandelbrot-Programm, bei dem die entsprechenden Arithmetikroutinen an den 68020 angepaßt wurden.

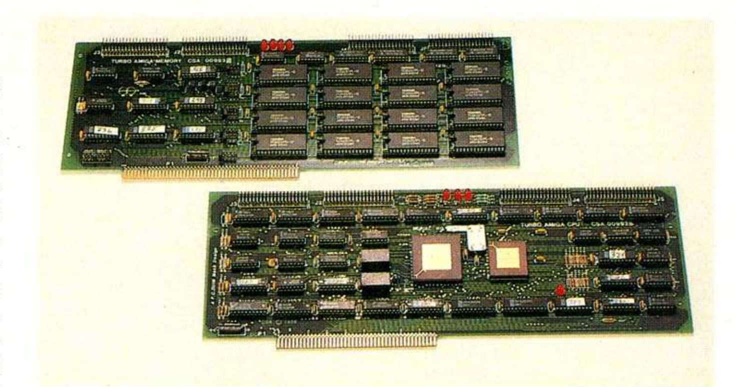


Bild 7. Diese beiden Karten machen aus dem Amiga 2000 einen neuen Computer mit erstaunlicher Geschwindigkeit

Probleme bereitet einzig der MOVE,SR-Befehl, der beim 68020 nur im Supervisor-Modus verwendet werden kann. Dieser Fehler taucht aber nur bei wenigen Programmen auf, so daß die Turbo-Karte relativ kompatibel ist (ein Programm, das diesen Fehler behebt, wird laut Hersteller mitgeliefert).

Um optimale Ergebnisse zu erhalten, empfiehlt es sich, RAM-Karten einzubauen, deren Adreßbus 32 Bit breit ist, damit die 32-Bit-Daten ohne Geschwindigkeitsverluste übermittelt werden können. Die Karten sind in verschiedenen Ausführungen mit 512 KByte, 4 MByte und 8 MByte erhältlich. Da bei solchen Dimensionen natürlich auch größere Externspeicher nötig werden, bietet GIT einen SCSI-Controller zum Preis von 1395 Mark an, der mit vier DMA-Kanälen arbeitet (davon sind drei für spätere Anwendungen frei) und an den maximal sieben SCSI-Festplatten angeschlossen werden können.

Der Turbo ist damit ein Weg, den A2000 (etwa in Verbindung mit der angekündigten Grafikkarte — 1280 x 1024 Pixel, zwei Millionen Farben) zu einer professionellen Workstation zu machen. (dm)

CPU/020/881A/14MHz 3100 Mark
CPU/020/881B/20MHz 4995 Mark
SRAM 512 KByte RAM 3525 Mark
SCSI-Controller 1395 Mark
BIG BYTE 4 MByte RAM 3995 Mark
BIG BYTE 8 MByte RAM 6500 Mark
GIT, Maassenstr. 10, 4235 Schermbeck, Tel. (02853) 4129

AT-Power

Commodore stellte endgültig die AT-Karte für den A2000 vor: Sie wird, wie bereits die PC-/XT-Karte, als (Janus-) Brückenkarte in jeweils einen PC-/AT- und einen Amiga-Slot gesteckt. Der »AT« besitzt einen mit 8 MHz getakteten 80286-Prozessor und 512 KByte Hauptspeicher. Ein Sockel für einen Arithmetik-Coprozessor (80287) ist vorhanden, ebenso ein Anschluß für ein zweites Laufwerk. Laut Angaben von Commodore ist der »Zusatzcomputer« 100prozentig kompatibel zu allen auf dem Markt erhältlichen PC-/XT-/AT-Modellen. Das 1995 Mark teure AT-Set besteht aus der AT-Karte, einem 5¼-Zoll-1,2-MByte-Diskettenlaufwerk und der dazugehörigen Software (MS-DOS 3.2/GW-Basic). Sobald uns ein Testmuster zur Verfügung steht, werden wir für Sie einen ausführlichen Testbericht in einer der nächsten Ausgaben bringen. (dm)

Commodores Hardwareküche

Der neue Leiter der Entwicklungsabteilung bei Commodore Braunschweig ist Dieter Preiss. Wir haben ihn besucht, um Einblick hinter die Kulissen der Geburtsstätte des Amiga 2000 zu nehmen.

Die »Offene Systemarchitektur« macht den Amiga 2000 zu einem der interessantesten Computer dieser Tage. Daß er aus dem Commodore-Werk Braunschweig kommt, weiß inzwischen fast jeder. Kaum jemand kennt jedoch die Entwickler. Einer davon ist der inzwischen zum Leiter der Abteilung aufgestiegene Dieter Preiss, der maßgeblich am Gesamtkonzept und insbesondere an der Programmierung der System-Software beteiligt war. Dem Kernteam von zwei Hardware- und fünf Softwareentwicklern wurde Anfang 1986 die Aufgabe gestellt, einen neuen Amiga zu entwerfen, der kompatibel zum alten Modell sein sollte. Das Gerät mußte ein überlappendes Bus-System haben und die Möglichkeit bieten, andere CPU's zu integrieren. Dieter Preiss dazu:



Bild 1. Dieter Preiss, der neue Leiter der Commodore Entwicklungsabteilung

Preiss: »Unsere Studie war das Sidecar für den Amiga 1000. Von der Architektur her ist der Amiga 2000 sehr ähnlich aufgebaut.«

Ein anderer Prozessor, wie zum Beispiel der Intel 8088 auf der PC-Karte, wird im Amiga 2000 so integriert, daß er, könnte so ein Chip denken, der Meinung wäre, er befände sich in einem ganz normalen Personal Computer. Der Amiga gaukelt ihm die Umgebung vor, die er zum Funktionieren benötigt.

Manche behaupten, der PC-Teil im Amiga sei zu langsam. Das entspricht natürlich nicht den Tatsachen. Das einzige, was das Erscheinen der PC-Informationen auf dem Amiga-Bildschirm langsam macht, ist die Umwandlung der PC-Grafik in das Amiga-Format.

Preiss: »IBM hat schon ein seltsames Grafik-Format. Es ist nicht einfach, die Bytes für die Bitplanes neu zu sortieren.«



Bild 2. Die Platinen des Amiga 2000 in der Einbrennphase

Dieter Preiss, der seit sechs Jahren bei Commodore arbeitet, muß es wissen. Viel aus den Programmen, die sich auf der Workbench befinden und den PC-Teil mit dem Amiga verbinden, ist unter seiner Regie entstanden.

Preiss: »Es war schon eine tolle Arbeitsatmosphäre. Wir haben Nächte durchgearbeitet, denn keiner wollte aus seinen Gedanken raus. So ist zum Beispiel die technische Struktur inklusive Blockdiagramm in nur drei Wochen entstanden. Die Programmierer waren wie Künstler, die nur in einem Team das Ganze fertigbringen konnten. Es kam auf die enge Kooperation von Hardware-, Amiga- und PC-Spezialisten an.«

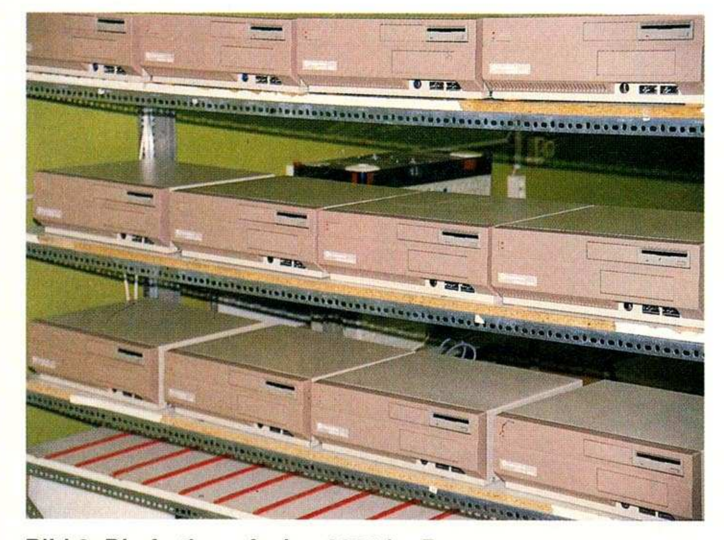


Bild 3. Die fertigen Amiga 2000 im Dauertest

Auf die Frage nach technischen Problemen der PC-Karte und Lieferschwierigkeiten des Amiga 2000 allgemein, war eine erstaunliche Antwort zu hören.

Preiss: »Immer gibt es Verbesserungen, aber keine einschneidenden Änderungen. An der Software wurde sorgfältigst gearbeitet; sie darf nicht krank sein. Eine Hardware kann man immer reparieren. Die Kinderkrankheiten sind jedoch ausgemerzt. Unser eigentliches Problem ist jetzt nur die zu große Nachfrage. Unweigerlich kommt es zu Lieferengpässen.«

Dabei wurden insgesamt schon 100000 Amiga-Modelle in Deutschland an die Händler ausgeliefert. Das ließ Commodore anläßlich der Systems in einer Presseverlautbarung nicht ohne Stolz verlauten.

Der studierte Physiker Preiss kennt sich mit seinen Erfahrungen aus den alten Zeiten des 8032-Prozessors natürlich inzwischen auf dem Amiga glänzend aus.

Preiss: »Ein solches Spezialwissen bekommt man nirgends gelehrt. Man muß es sich selbst erarbeiten und alle Fehler machen. Wie man an eine CPU herangeht, die eigentlich sehr dumm ist, kann man nur durch Erfahrung lernen. Man muß sich durchbeißen.«



Bild 4. Entwickler Burgdorff vor den Paletten der Tagesproduktion an Amiga 500

Nach einem Rundgang durch das Braunschweiger Werk, in dem auch Amiga 500, PC 1 oder C 64 vom Band laufen, sind noch interessante Neuheiten zu erwähnen. So ist unter anderem ein neues Amiga-DOS in der Planung, das schneller arbeiten und dessen Testphase bis Jahresende abgeschlossen sein soll.

Preiss: »Man könnte im Prinzip ein weiteres DOS nachladen und zusätzlich mit dem alten verwenden.«

Nur das Booten von Festplatte bereitet den Entwicklern noch einiges Kopfzerbrechen. Dazu müßte nämlich aus dem fast völlig ausgelasteten Kickstart-ROM etwas herausgenommen werden, um Platz für die Harddisk-Routinen zu schaffen. Aber auch in dieser Richtung sind schon Entwicklungen im Gange. (jk)

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Albert Absmeier (aa)
Stellvertr. Chefredakteur: Georg Klinge (gk)
Redaktion: Peter Aurich (pa), René Beaupoil (rb), Ulrich Brieden (ub), Jörg Kähler (jk), Dieter Mayer (dm)

Hotline: Gerd Donaubauer (640), Monika Welzel (640)

Redaktionsassistenz: Monika Lewandowski (222), Andrea Kaltenhauser (202)

Fotografie: Jens Jancke, Claudia Kränzle

Titelgestaltung: Heinz Rauner, Grafik Design, Werner Nienstedt

Layout: Leo Eder (Ltg.), Rolf Raß (Cheflayouter), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-415656, Telex: 862329 mut ch
USA: M & T Publishing, Inc: 501 Galveston Drive, Redwood City, CA 94063; Tel. (415) 366-3600, Telex 752-351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Produktionsleitung: Klaus Buck

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenleitung: Alicia Clees (313)

Anzeigenverwaltung und Disposition: Patricia Schiede (172), Monika Burseg (147)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigen-Auslandsvertretungen:
England: F. A. Smyth & Associates Limited 23a, Aylmer Parade, London, N2 OPQ, Telefon: 0044/1/3405058, Telefax: 0044/1/3419602
Taiwan: Third Wave Publishing Corp. 1 — 4 Fl. 977 Min Shen E. Road, Taipei 10581, Taiwan, R.O.C., Telefon: 00886/2/7630052, Telefax: 00886/2/7658767, Telex: 078529335

Vertriebsleiter: Helmut Grünfeldt (189)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement kann jederzeit zum Ende des bezahlten Zeitraums gekündigt werden.

Bezugspreise: Das Einzelheft kostet DM 7,—. Der Abonnementspreis beträgt im Inland DM 79,— pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,— für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z. B. USA) um DM 38,—, in Ländergruppe 2 (z. B. Hongkong) um DM 50,—, in Ländergruppe 3 (z. B. Australien) um DM 68,—.

Druck: R. Oldenbourg GmbH, Hürderstr. 4, 8011 Kirchheim

Verantwortlich: Für redaktionellen Teil: Albert Absmeier. Für Anzeigen: Alicia Clees.

Redaktions-Direktor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Start ohne Icon?

Kann man von der Workbench aus, ohne Aufruf des CLI, ein Programm, das kein Icon besitzt, starten?
PATRICK SPEIDEL

Gemischtes Doppel

Ich besitze den Amiga 500 und den 128D sowie den MPS 2000 C-Farbdrucker. Beide Computer möchte ich am Drucker anschließen. Kann mir jemand einen Tip geben?
WERNER SCHNITZLER

Verhüterli

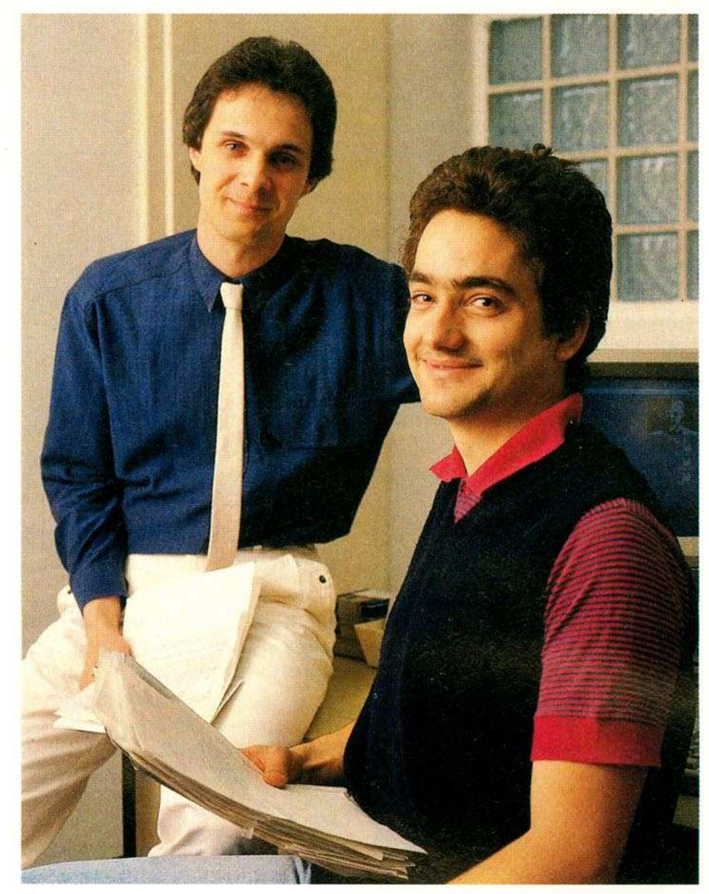
Besonders im Zusammenhang mit Computerviren interessiert mich eine Frage: Gibt es eine Softwaremöglichkeit, den Disketten-Schreibschutz zu umgehen? Kann ich bei schreibgeschützten Disketten wirklich ganz beruhigt sein?
C. BECKER

Apostroph gesucht

Bei meinem Amiga 500 funktioniert die Apostroph-Taste, ich meine die Taste rechts über dem »ü«, im Zusammenhang mit der deutschen Tastaturbelegung nicht. Da mit anderen Belegungen (zum Beispiel USA) dieser Fehler nicht auftritt, nehme ich an, daß dies an dem Tastaturtreiber selbst liegt. Kennt jemand dieses Problem und kann mir sagen, wie man an das Apostroph herankommt?
ERNST RAUPP, Ausgabe 10

Die Apostroph-Taste rechts oberhalb des »ü« ist eine sogenannte »Tottaste«, die es auch auf vielen anderen Computern und guten Schreibmaschinen gibt. Wird sie alleine gedrückt, so bewirkt dies nichts. Erst wenn Sie danach zum Beispiel ein kleines »a« tippen erscheint »á« (Accent Aigue).

Die Apostroph-Taste auf dem Amiga 500 dient also eigentlich dazu, Akzente zu setzen. Dies ist für Fremdsprachen wichtig. Betätigen Sie nach der vermeintlichen Apostroph-Taste die Leertaste, so erhalten Sie — scheinbar — auch das gesuchte Apostroph auf dem Bildschirm. Dies ist aber nicht das Zeichen, das Programmierer benötigen, um in Basic Kommentare zu kennzeichnen. Auch in C-Programmen benötigen Sie ein anderes Zeichen. Des Rätsels Lösung: Das gesuchte Apostroph versteckt sich hinter der Tastenkombination <ALT ä>. Überhaupt gelangt man mit <ALT> und anderen Tasten an viele aufregende Zeichen und Buchstaben. Auch einige weitere Akzente finden Sie in verborgenen Tottasten. Ausprobieren lohnt sich.
EDMUND OTT



Unsere Redakteure Ulrich Brieden und Jörg Kähler wählen die Beiträge für das Leserforum aus

Basic-Autostart

Ist es möglich, ein Basic-Programm mit Autostart in die »Startup-sequence« einzubinden, welches dann seinerseits den Amiga-Basic-Interpreter lädt und anschließend auch startet?
HAJO SCHNEIDER

Einen Autostart von Basic-Programmen erreichen Sie, indem Sie die »Startup-Sequence« Ihrer Bootdiskette ändern. Um ein bestimmtes Programm zu laden, genügt eine Zeile, die sowohl das Programm als auch das Amiga-Basic lädt:

```
RUN amigabasic programm-
name
```

(ub)

Basic oder C?

Der Leserbrief von Herrn Hilpert in Ausgabe 10 hat eine starke Reaktion hervorgerufen. Viele Leser der AMIGA haben uns ihre Einstellung zu der Frage geschrieben, ob und in welchem Umfang wir C-Programme veröffentlichen sollen. Einige Meinungen möchten wir zur Diskussion stellen:

Mehr C

In der Ausgabe 10/87 hat ein gewisser Martin Hilpert etwas zu C- und Assemblerprogrammen geschrieben, mit dem ich mich als Amiga-Programmierer ganz und gar nicht einverstanden erklären kann. In dieser ansonsten sehr guten Zeitschrift (»Vielen Dank«, die Redaktion) müßten noch viel mehr C- und Assemblerlistings zu finden sein. Für C und Assembler wurden die Reference-Manuals geschrieben, nicht für Basic. Nur mit C und Assembler kann ein Programmierer die Systemroutinen des Amiga nutzen. Nur mit ihnen lassen sich zeitkritische Routinen realisieren. Sicher, jeder hat Amiga-Basic, aber damit läßt sich nichts dergleichen machen.
PETER M. HEGEMANN

C ist wichtig

Ich fände es sehr schade, wenn die momentan sehr wichtige und beliebte Sprache C aufgrund der von Herrn Hilpert angegebenen Einwände von Ihnen nicht mehr unterstützt werden sollte. C bietet wesentlich effektivere Problemlösungen als das zugegebenermaßen auch nicht schlechte Amiga-Basic. Viele meiner Bekannten, die mit dem Amiga liebäugeln, interessiert jedoch besonders der Einsatz anderer Programmiersprachen. Nichts gegen Basic-Listings; einfache Probleme sind in Basic häufig schneller zu lösen. Daher mein Vorschlag: Bleiben Sie bei einer ausgewogenen Mischung.
JAN PETER MOHR

Zuviel C

Ich bin ebenfalls der Meinung, daß zu viele C-Programme veröffentlicht werden. Da viele Einsteiger sich keinen C-Compiler leisten können, sind diese Listings unbrauchbar. Zumindest so lange, bis sie nicht in reiner Maschinensprache abgedruckt werden, wenn möglich mit Prüfsummen wie beim MSE. Auch für Basic-Programme wären Prüfsummen sehr wünschenswert. Nach drei Jahren »64'er« ist ein Abtippen ohne Kontrolle frustrierend.
FRANZ SAGMÜLLER

Checksummer

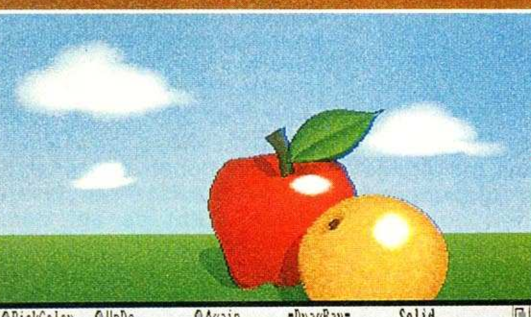
Die Frustration beim Abtippen entfällt erfreulicherweise schon mit der Ausgabe 12 des Amiga-Magazins, die Sie gerade in den Händen halten. Wir haben die guten Erfahrungen mit dem Checksummer aus der 64'er bereits umgesetzt. Sie finden den Checksummer zum Amiga auf Seite 76.

Die entfachte Diskussion um die Programmiersprachen werden wir weiter verfolgen. Sie sehen an den drei oberen Beispielen, wie groß das Spektrum unterschiedlicher Meinungen ist. Wir möchten natürlich die Wünsche aller unserer Leser berücksichtigen. Um die unterschiedlichen Interessen zu kennen, brauchen wir Ihre Hilfe. Schreiben Sie uns, aus welchen Gründen Sie eine bestimmte Programmiersprache bevorzugen, und welche Listings Sie mehr interessieren. Wir haben nicht vor, total auf C-Programme oder Basic-Listings zu verzichten. Es gilt einen Kompromiß zu finden. Ihre Vorschläge sind gefragt. (ub)

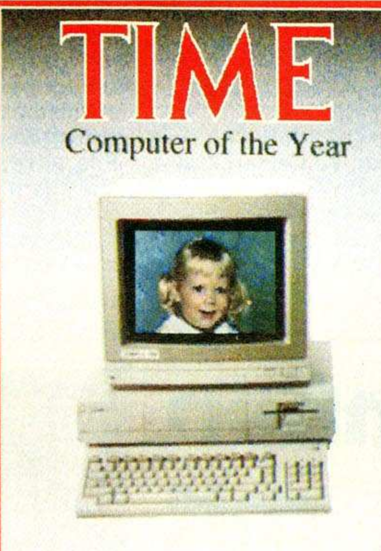




DIGI PAINT







Das erste Malprogramm, das die 4.096 Farben des Amigas voll ausnutzt.
Jetzt gibt es Digi-Paint in einer deutschen Version.

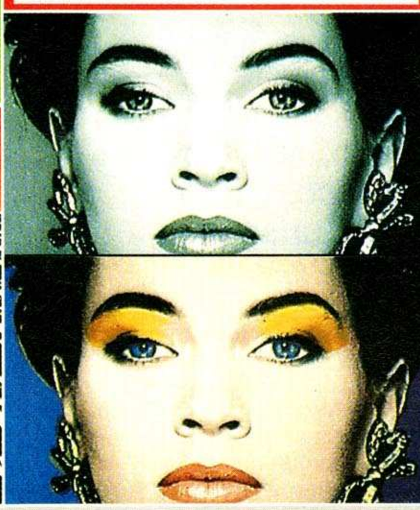
Die Vorteile von Digi-Paint sind:

- voll IFF und Digi-View kompatibel
- 4.096 Farben zur gleichen Zeit
- exaktes Ausschneiden von Bildteilen und Wiedereinsetzen an anderer Stelle
- komplett in Assembler programmiert
- einfache Bedienung von Farben und Menüleiste
- volles Abspeichern und Laden von IFF-Bildern
- zwölf verschiedene Malmoden einschließlich Vermischen, Färben und Schattieren
- Kreise, Elipsen, Rechtecke und andere Zeichenhilfen
- es arbeitet im 320 x 200 und im hi-res 320 x 400 HAM-Modus
- zum detaillierten Arbeiten kann man Ausschnitte vergrößern
- Effekte wie Verdoppeln, Halbieren, Spiegeln und mehr

Preis DM **138,–**







Alle hier gezeigten Bilder wurden mit Digi-Paint erstellt. Digi-Paint ist natürlich Digi-View-kompatibel.

Die deutschen Digi-Paint- und Digi-View-Pal-Versionen erhalten Sie von Ihrem Commodore-Händler oder direkt von ATLANTIS bzw. COMPUSTORE.





Soft- und Hardware GmbH
Dunantstraße 53 · 5030 Hürth
Telefon 0 22 33 / 4 10 81

&

COMPUSTORE
Handelsgesellschaft mbH für Hard- und Software
Fritz-Reuter-Straße 6 · 6000 Frankfurt/Main 1
Telefon 069/567399

Polydraw in Assembler?

In Ausgabe 6/7 des Amiga-Magazins habe ich mit Interesse den Artikel auf Seite 76 über den Polydraw-Befehl des Betriebssystems gelesen. Was mich nun interessiert, ist: Wie erhält man in Assembler den »RastPort«-Wert? Ich hoffe, daß mir jemand weiterhelfen kann.
INGMAR REYER

Der Rastport ist der Name einer Datenstruktur, in der sich grafische Daten eines Fensters oder Screens befinden. Mit der Startadresse des Rastport übergeben Sie den Zeichenfunktionen der »graphics.library« wichtige Werte, beispielsweise die Startadresse der Bitplanes, die Größe der Zeichenfläche oder die Farbe, mit der gezeichnet werden soll. Unter Intuition besitzen sowohl Screens als auch Windows ihren eigenen Rastport. In der Fensterstruktur befindet sich ein entsprechender Zeiger. Laden Sie zunächst den Zeiger auf die Windowstruktur in ein Adreßregister. Beispiel:

```
move.l mywindow,a0
```

Den Pointer auf den Rastport erhalten Sie durch indirekte Adressierung mit Distanz. Der entsprechende Offset in der Window-Struktur beträgt 50:

```
move.l #50(a0),a0.
```

Der Rastport eines Screens ist Bestandteil der Screenstruktur. Sie erhalten den Zeiger, indem Sie 80 zum Zeiger auf die Screendaten addieren.

```
move.l myscreen, a0
adda # 80 ,a0
```

Im Register a0 befindet sich die Adresse des Rastport.

(ub)

Amiga Assembler-Paket

Da ich zur Zeit einen Zwei-Pass-Assembler auf dem Amiga schreiben möchte, stellen sich mir folgende wichtige Fragen:

1. Welche Struktur hat der vom Systemassembler »ASSEM« erzeugte linkbare Code (damit die Programme kompatibel bleiben)?
2. Welche Standard-Pseudoopcodes empfehlen sich für den 68000-Prozessor? Welche Funktionen haben die Pseudoopcodes »STRUCT« und »APTR«?
3. Wie sieht die Struktur des Codes aus, der von den Linkern »ALINK« beziehungsweise »BLINK« erzeugt wird?
CARSTEN MIELKE

File schon geöffnet

Obwohl nach Erstellen eines Programms mit Amiga-Basic dieses ordnungsgemäß gespeichert wurde, erscheint bei erneutem Arbeiten mit diesem Programm bei mir oft die Fehlermeldung »File already open«. Besonders gern erscheint diese Meldung, wenn die Datei mit »LLIST« ausgedruckt werden soll (mit Drucker Commodore MPS 2000 C). Um den Fehler zu vermeiden, muß das System jedesmal neu gebootet werden, was natürlich sehr umständlich ist. Wie kann man diesen Ärger vermeiden? DR. T. REICHELT

Amiga 2000 mit neuer Platine

Mein Amiga 2000 besitzt eine ganz neue Hauptplatine mit der Bezeichnung »B2000-CR/Rev 4.1«. Auf der Platine befindet sich der gesamte Speicher von 1 MByte. Es gibt keine Erweiterungskarte mehr. Der alte Amiga 2000 läßt sich, wie in der AMIGA 8/9 beschrieben, einfach und preiswert auf 1,5 MByte erweitern. Gibt es auf dem 2000B hierzu auch eine Möglichkeit? Wie kann ich ein zweites Laufwerk in den neuen Amiga 2000 einbauen? Gibt es eventuell noch andere Unterschiede zum älteren Modell?
GÜNTER DE HEUVEL

Gewinnen mit Musik

Haben Sie musikalische Ideen, die nur darauf warten, an die Öffentlichkeit zu gelangen? Machen Sie mit bei unserem Musikwettbewerb; tolle Preise winken.

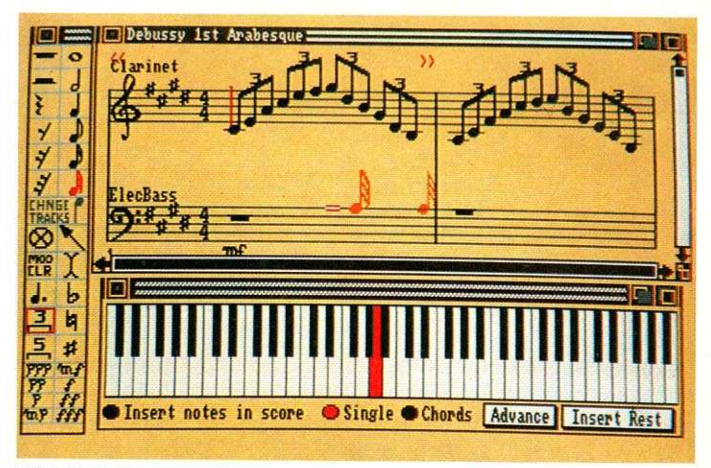


Bild 1. Sehr gut wäre es, wenn die Songs als IFF-Files vorliegen würden; das erleichtert die Auswertung.

Viele der heutigen Profis auf dem Gebiet Computermusik haben auch mal klein angefangen. Auch sie mußten zuerst lernen, wie man Töne so zusammenfügt, daß Melodien herauskommen, die das menschliche Ohr erfreuen. Jedoch ist noch kein Meister vom Himmel gefallen. Auf dem Amiga mit seinen guten Editorprogrammen, wie beispielsweise »Deluxe-Music« (Bild 1) oder »Sonix«, hat man es heute allerdings ungleich leichter, Ideen umzusetzen. Wir suchen originelle Musikstücke, die mit einem für den Amiga käuflichen Programm erstellt wurden. Die verwendeten Instrumente können Sie selbst kreieren oder auch digitalisierte Sounds einbinden. Vermerken Sie bitte in einem beigelegten Schreiben, welches Musikprogramm Sie verwendet haben und welche Instrumente in welchen Unterverzeichnissen vorliegen müssen, damit der Song sofort mit dem jeweiligen Editor geladen und abgehört werden kann. Im wesentlichen kommt es auf die Originalität der Melodie an. Es hat also wenig Sinn, bekannte Lieder umzusetzen oder Teile anderer Hits zu verwenden. Natürlich sollte das Lied auch gut klingen; hauptsächliches Bewertungskriterium ist jedoch die Neuheit und Schwierigkeit des verwendeten Notenmaterials. Bitte vergessen Sie nicht, eine unterschriebene Copyright-Erklärung beizufügen, daß Sie das eingesandte Material selbst komponiert und programmiert haben.

Es gibt folgende Preise zu gewinnen:

1. und 2. Preis: Eine Reise nach Frankfurt in die Offenbacher Master Studios. In diesem mit High-Tech vollgestopften Sound-Labor begann Michael Münzing mit dem Computer-Song »Changing Minds« seine Karriere. Ein Besuch bei Commodore Frankfurt ist ebenfalls Bestandteil des Programms.

3. bis 5. Preis: Das super Musikprogramm »Sonix« von Aegis für Ihren Amiga.

6. bis 20. Preis: Eine Maxi-Single mit dem Hit »Changing Minds« der Gruppe »16 Bit«.

Schicken Sie Ihr Musikstück mit Instrumenten auf Diskette und Ihrer Copyright-Erklärung an:

Markt & Technik Verlag AG
Redaktion Amiga
Musikwettbewerb
Hans-Pinsel-Straße 2
8013 Haar bei München

Einsendeschluß ist der 31.12.1987

Ausgeschlossen sind Mitarbeiter des Verlages und deren Angehörige.

(jk)

Festplatten —

Jeder kennt den Begriff Festplatten oder Hard-Disks. Doch was genau bedeuten sie? Lesen Sie, wie solch ein Gerät aufgebaut ist und wie es funktioniert.

Wenn Disketten zur Datenspeicherung aus Platzgründen nicht mehr genügen, müssen leistungsstärkere Speichermedien her. Ein preiswerter Weg, zu viel Speicher zu kommen, sind Festplattenlaufwerke, auch gerne als Hard-Disks bezeichnet. Sie bewältigen Datenmengen von 10, 20, 40, 80 und mehr MByte. Sie erlauben schnelle Zugriffe auf Daten und sind doch manchmal nicht größer als ein Diskettenlaufwerk. Wie diese hochkomplizierten technischen Errungenschaften aufgebaut sind und wie sie funktionieren, bringt Ihnen der folgende Artikel näher.

Masse wird erschwinglich

Bei modernen Computern wie etwa Personal Computern oder dem Amiga werden immer mehr Speichermedien gefragt, die hohe Datenmengen bewältigen können und zudem auch schnell sind. Aus diesem Grund haben sich die Festplatten durchgesetzt.

Die starren Platten in den Gehäusen sind zwar nicht mehr wie bei Diskettenlaufwerken auswechselbar, dafür ist die Kapazität, wie bereits erwähnt, sehr hoch. Solange keine großen Anwendungen wie beispielsweise Vernetzungen (Mehrplatzsysteme) das Vorhandensein einer »riesigen« Platte erforderlich machen, kommt man bei kleineren und mittelgroßen Computern voll mit gebräuchlichen Hard-Disks aus. Nicht zuletzt, da der Preis in erschwingliche Regionen gerutscht ist — eine 20-MByte-Filecard bekommt man schon für 800 Mark — finden diese Geräte immer mehr Einsatz auch bei Heim- und Personal Computern.

Im Gegensatz zu Diskettenlaufwerken ist ihre Zugriffsgeschwindigkeit (Zeit, um Daten zu lesen) um ein Vielfaches schneller. Dies liegt daran, daß die Platten ständig rotieren, während Floppylaufwerke erst anlaufen müssen. Ein gebräuchliches Floppylaufwerk hat eine durchschnittliche Zugriffszeit von etwa 200 bis 250 Millisekunden, Festplatten liegen zwischen 20 und 100 Millisekunden.

Während Disketten, die mit Amiga-DOS formatiert wurden, »nur« 80 Spuren je Diskettenseite, 11 Sektoren je Spur und 512 Byte je Sektor, also 80 x 11 x 2 x 512 = 901 120 Byte fassen (siehe auch Schema in Bild 1), liegt die Zahl der nutzbaren Bytes bei einer Platte viel höher. Eine gängige 20-MByte-Hard-Disk wird mit 612 Spuren und 17 Sektoren je Plattenseite formatiert. Sie besitzt vier Schreib-/Leseköpfe und zwei Platten (612 x 17 x 4 x 512 = 21 307 392 entspricht etwa 20 MByte). Je nach verwendeter Hard-Disk können auch mehr Platten und Schreib-/Leseköpfe verwendet werden. Auch kann die Zahl der Spuren und Sektoren variieren. In Bild 2 sehen Sie einen Querschnitt durch ein größeres Festplattenlaufwerk. Deutlich zu sehen sind die einzelnen Platten und die Schreib-/Leseköpfe. Bild 3 zeigt dasselbe Schema noch einmal, diesmal aber mit einer echten, aufgeschnittenen Hard-Disk.

Leider bringt die große Kapazität eine (theoretisch) hohe Störanfälligkeit mit sich. Durch die Schreibdichte von bis zu 2000 Spuren je Inch (2000 tpi) hat eine Hard-Disk viel kleinere Schreib-/Leseköpfe als etwa ein Diskettenlaufwerk, das normalerweise mit maximal 135 Spuren je Inch (135 tpi) arbeitet. Die hohe Schreibdichte kann vor allem deshalb erreicht werden, weil die Platten fest auf einer Spindel fixiert sind und so der Kopf ohne Spiel positioniert werden kann. Der Festplatten-Schreib-/Lesekopf liegt auch nicht wie bei Diskettenlaufwerken auf der magnetisierten Oberfläche der Platte auf, sondern schwebt von einem Luftpolster getragen einen Mikrometer ($^1/_{1000}$ Millimeter) über der Platte. Das Luftpolster wird durch die Plattenumdrehung (3600 Umdrehungen pro Minute) erzeugt. Schon der kleinste Stoß an ein arbeitendes Plattenlaufwerk kann verheerende Folgen haben: Der Schreib-/Lesekopf gerät in Schwingungen und kann die rotierende Scheibe berühren. Durch die hohe Plattendrehzahl könnte sich der Kopf richtig in die Plattenoberfläche »einfräsen«, womit natürlich alle gespeicherten Daten unlesbar sind und die Hard-Disk kaputt ist. Dieser unangenehme Vorgang wird in der Fachsprache als »Head-Crash« bezeichnet. Aber auch Staub kann zerstörende Folgen haben, wenn er in das gekapselte und normalerweise staubfreie Gehäuse eindringen sollte. Schon kleinste Rauchpartikel im Zigaretten-

Diskettenhülle
Schreibschutz
Sektor 1
Sektor 2
Sektor 3
Spur 78
Spur 79
Spur 2
Spur 1
Magnetscheibe

Bild 1. Dieses Bild zeigt wie eine unter Amiga-DOS formatierte Diskette in Spuren und Sektoren aufgeteilt ist



Bild 3. Und so sieht's wirklich aus: ein aufgeschnittenes Festplattenlaufwerk mit drei einzelnen Platten Foto: BASF

Was ist das?

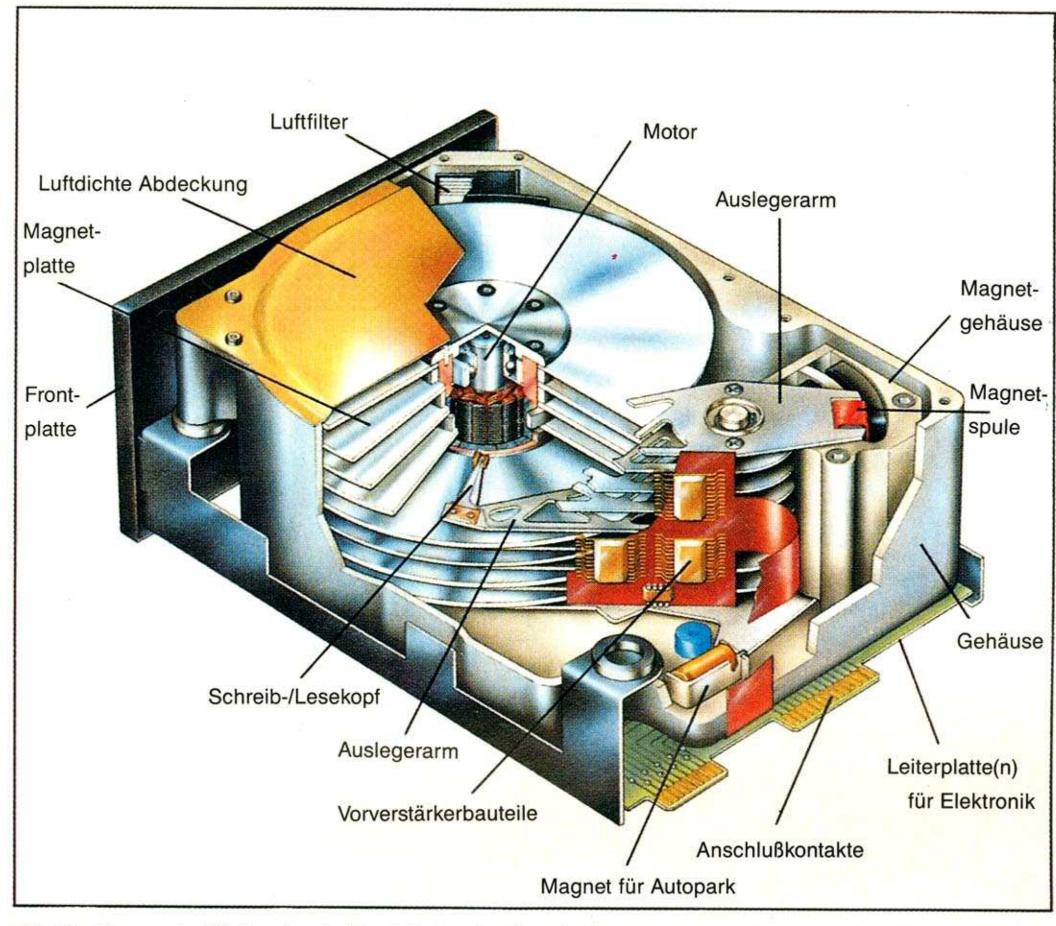


Bild 2. Querschnitt durch ein Festplattenlaufwerk. Deutlich sind der Plattenstapel, der Luftfilter gegen Verunreinigungen und die Schreib-/Leseköpfe zu sehen.

rauch erreichen eine Größe von zwei Mikrometer, ein Menschenhaar (Durchmesser 50 Mikrometer) wäre für den Schreib-/Lesekopf wie eine Felswand.

Deshalb sind Plattenlaufwerke staubdicht gekapselt. Der Zusammenbau der Geräte erfolgt in speziellen, staubfreien Räumen. Außerdem enthalten die Laufwerke in ihrem Inneren zusätzlich einen Staubfilter, der Verunreinigungen im Gehäuse selbst ausfiltern soll (Bild 2). Eine weitere Sicherheitsmaßnahme, um Beschädigungen zu vermeiden, ist, die Schreib-/Leseköpfe bei Nichtbenutzung in eine Parkposition zu fahren, in der sie vor Erschütterungen sicher sind. Neuere Platten erledigen dies automatisch nach dem Abschalten des Geräts (Autopark), während manche älteren Platten noch »von Hand« (durch Software) arretiert werden müssen. Durch all diese Maßnahmen sind Hard-Disks heutiger Fertigung relativ sicher vor Beschädigungen oder Datenverlust, insofern sie nicht schweren Belastungen ausgesetzt werden.

Sicher haben auch Sie sich schon einmal Gedanken darüber gemacht, ob es nicht vielleicht doch einmal in Frage kommen sollte, sich eine Hard-Disk zuzulegen. Aber worauf kommt es beim Kauf an?

Kapazität für die Zukunft

Das ausschlaggebende Motiv zur Entscheidung für eine bestimmte Platte ist sicher deren Speicherkapazität. Kapazitäten von 5 oder 10 MByte, wie sie vor einigen Jahren gefertigt wurden, sind heutzutage fast nicht mehr am Markt zu finden. Normalerweise fängt die untere Grenze bei 20 MByte an. Der Kapazität nach oben sind fast keine Grenzen gesetzt. Die größte uns bekannte Platte verfügt beispielsweise über das satte Speichervolumen von über 1,1 GByte (1 Gigabyte = 1000 MByte). Solche Größen kommen natürlich für einen durchschnittlichen Anwender nicht in Frage.

Sollte die Kapazität von 20 MByte nicht ausreichen, läßt sich in den Amiga auch eine Platte mit mehr »Volumen« einbauen beziehungsweise anbauen. Größen von 33, 40, 50 oder 80 MByte werden bereits im Fachhandel angeboten.

Aber nicht nur die Kapazität ist entscheidend. Gewichtige Gründe zum Kauf einer Hard-Disk sind auch die Zugriffszeiten. Das bedeutet, die Zeit, die durchschnittlich benötigt wird, um einen Sektor anzufahren und die Daten zu lesen. Ein Mittelwert für 20- und 30-MByte-Laufwerke ist etwa 75 bis 85 Millisekunden. Langsamere Plattenstationen werden heutzutage auch fast nicht mehr gefertigt.

Besitzt das Gerät »Autopark«? Werden also die Schreib-/Leseköpfe automatisch beim Ausschalten in eine sichere Ruheposition gefahren? Dies ist ein nicht zu unterschätzender Vorteil, um Beschädigungen etwa während eines Transports zu vermeiden (wie leicht vergißt man, die Köpfe »per Hand« zu parken).

Die Größe der gesamten Hard-Disk sollte auch nicht unterschätzt werden. Hat man wenig Platz auf dem Arbeitstisch oder im Gehäuse zur Verfügung, sollte ein kleineres Gerät gekauft werden. Standardgrößen für Plattenlaufwerke sind 3½- und 5¼-Zoll. Die kleineren Festplatten besitzen heutzutage Kapazitäten von 40 MByte, sollten also bei Platzfragen wegen ihrer geringeren Größe ausgewählt werden.

Platz für viele Daten

Leider sind auf dem Markt derzeit noch relativ wenig Festplattenstationen für den Amiga 500/1000 vorhanden, da erst spezielle Controller zur Ansteuerung der Geräte entwickelt werden müssen. Für den Amiga 2000 sieht es da schon besser aus. Commodore bietet seit Anfang Oktober einen Controller an, der es erlaubt, beliebige Plattenlaufwerke anzuschließen. Der Controller bewältigt Kapazitäten von mehreren GByte, so daß wohl keine Probleme wegen zu geringer »Speicherfreiheit« auftauchen werden. Außerdem können in den A2000, sofern er über eine PC-/AT-Karte verfügt, verschiedene PC-Festplattencontroller eingebaut werden. Die maximale Kapazität ist hierbei nicht so hoch wie beim Commodore-Controller, doch dürften über 100 MByte »Volumen« genügen. Auch an diese Controller lassen sich die verschiedensten Plattenlaufwerke anschließen, so daß zumindest für den Amiga 2000 keine Knappheit an Festplatten besteht. Einzig die Baugröße dürfte hier Probleme bereiten. Geeignet für die PC-/AT-Seite des A2000 sind Filecards, da sie auch intern eingebaut werden können.

Welchen Amiga Sie auch immer besitzen und für welche Festplatte Sie sich auch entscheiden, auf jeden Fall ist es eine sinnvolle Investition, da Festplattenlaufwerke erheblich zum angenehmeren Arbeiten beitragen. (dm)



Die Grenzen sprengen . . .

Auch bei einem so leistungsfähigen Computer wie dem Amiga gibt es Dinge, die fehlen: Speicher, Diskettenlaufwerke oder Festplatten. Wie gut, daß es diese Accessoires im Fachhandel gibt.

Es ist schon einiges, was man beim Kauf eines Amiga für sein Geld bekommt: eingebautes Diskettenlaufwerk, 512 KByte RAM und ein leistungsfähiges Betriebssystem. Doch auch mit dieser Fülle stößt der Amiga-Benutzer schnell an Grenzen: Ein Diskettenlaufwerk ist zu wenig zum sinnvollen Arbeiten, die Programme fordern mehr Speicher und das Betriebssystem ist »unausgelastet«, wenn es nicht gefordert wird. Mit 1 MByte Hauptspeicher läßt sich dagegen schon einiges anfangen. Soll das Multitasking-Betriebssystem aber richtig genutzt werden, ist ein Speicherausbau auf 2 oder 4 MByte anzuraten. Wir geben Hilfestellung und sagen Ihnen, welche Erweiterungen für Sie in Frage kommen könnten.

Sicher kennen auch Sie das Ärgernis: Sie besitzen nur das eingebaute Diskettenlaufwerk und möchten mit anderen Disketten arbeiten. Aber kaum wird versucht, beispielsweise mit dem CLI (Command Line Interface) zu arbeiten, verlangt der Amiga das Einlegen der Startdiskette. Also Diskette wechseln, Befehl laden, neue Diskette einlegen...

Wie komfortabel ist es da doch, ein zweites Laufwerk angeschlossen zu haben. Die eine Diskettenstation enthält die Startdiskette, die andere die zu bearbeitende. Auf einmal entfällt das lästige Diskettenwechseln, man hat mehr Möglichkeiten zu Dateioperationen als mit nur einer Floppystation. Damit Sie einen Überblick über die auf dem Markt angebotenen Zweitlaufwerke erhalten, finden Sie in dieser Ausgabe eine Vorstellung und Übersicht empfehlenswerter Diskettenstationen zum Anschluß an den Amiga.

Einige Anwendungen benötigen für ihre Daten so viel Platz, daß die 880 KByte, die eine Diskette zur Verfügung stellt, nicht mehr ausreichen. Der Weg, dem zu entkommen, sind Festplattenlaufwerke. Sie bieten 20, 40, 70 oder mehr MByte Speicherkapazität und sind zudem ein Vielfaches schneller als die Diskettenlaufwerke. Harddisks sind inzwichen in Preisbereiche »gerutscht«, in denen sie auch für Interessenten mit kleinen Geldbörsen erschwinglich werden (teilweise unter 1500 Mark.) Sie finden in dieser Ausgabe eine Vorstellung von vier verschiedenen Festplatten, die sich zum Anschluß an die Amiga-Modelle eignen. Außerdem wird beschrieben, wie diese Geräte funktionieren und worauf es beim Kauf ankommt.

Für Bitkrabbler und Freaks ist der Artikel über die Diskettenstruktur das, wonach sie schon lange gesucht haben. Sie erfahren dort alles Wissenswerte über den Aufbau und die Struktur einer mit Amiga-DOS formatierten Diskette, welchen Sinn die einzelnen Bytes haben und wie das Betriebssystem anhand der einzelnen Linkpointer die Datenblöcke findet.

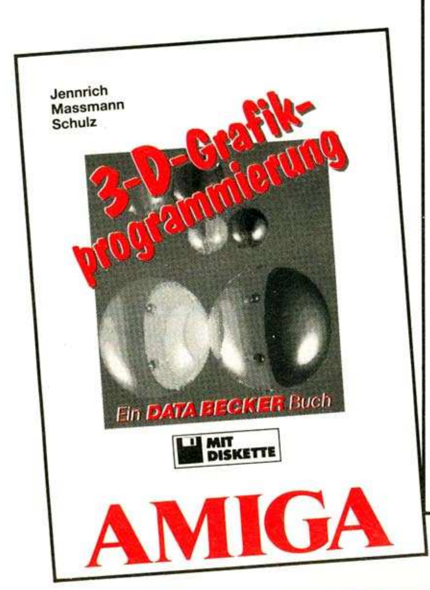
Das Programm des Monats, »DCopy«, paßt zu diesem Thema. Es ist ein Kopierprogramm, das gestattet, Diskettenkopien auf bequeme Art zu erstellen.

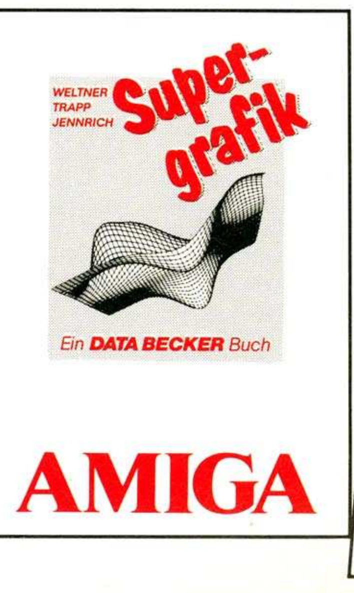
Zu einem Computer gehört auch die Software, die ihn erst zum Arbeiten befähigt. Sie finden Software-Tests über das »Futter« für Ihren Amiga. Seien es Grafik-Programme wie der »Animator:Apprentice« oder »Forms in Flight«. Besonders die Dateiverwaltung »Datamat« und das Textprogramm »Vizawrite« sind Programme, deren Tests Sie lesen sollten.

Programmierer bekommen im Listingteil wieder viele Programme zum Abtippen, die nutzbringend eingesetzt werden können. Nebenbei lernen Sie in unseren Kursen, wie Sie mit dem CLI umgehen, das Amiga-Basic programmieren oder mit C leistungsfähige Programme erstellen können. Wen interessiert, wie die Grafikhardware des Amiga Bilder auf den Schirm bringt, sollte sich unseren neuen Grafikkurs ansehen. Sie werden erstaunt sein, wieviel »Kraft« in diesen Mikrochips steckt.

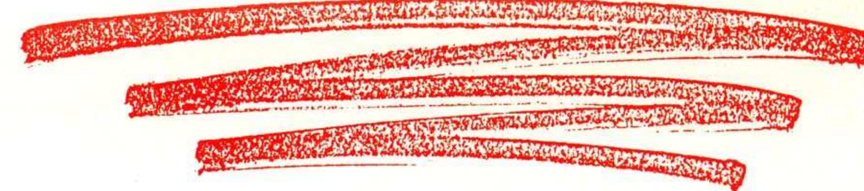
Selbstverständlich fehlt auch nicht die Public Domain-Seite, die Sie über interessante Freesoft-Disketten informiert. Freie Software ist eine Möglichkeit, mit wenig Geld zu guter Software zu kommen, damit der Amiga weiterhin das bleibt, was er ist: ein höchst leistungsfähiger Multicomputer. (dm)

Top aktuell:









3-D-Grafik auf dem Amiga – hier ist der Schlüssel zu dieser faszinierenden Welt. In diesem Buch werden Grafikalgorithmen beschrieben und erläutert, die es in sich haben. Mit ihnen können Sie absolut realistisch gestaltete Bilder erstellen: Die einzelnen Komponenten eines Bildes lassen sich dabei mit einem Editor problemlos eingeben und solange durch die Wahl verschiedener Materialien, Farben, Lichtquellen und Spiegelungen verfeinern, bis sie eine absolut naturgetreue Darstellung erreicht haben. Die Berechnung des Bildes übernimmt das Programm automatisch, in allen Auflösungen mit bis zu 4096 Farben. 3-D-Grafikprogrammierung – ein Muß für alle, die Spaß an tollen Grafiken haben.
Amiga 3-D-Grafikprogrammierung
Hardcover, 283 Seiten
inkl. Diskette, DM 59,–

Grafik ist das zentrale Thema des Amiga. Und mit dem Supergrafik-Buch können Sie hier in Dimensionen vorstoßen, die Sie selbst einem Amiga nicht zutrauen würden. Aber keine Sorge: Amiga Supergrafik wurde nicht nur für Profis geschrieben. Es bietet jedem etwas. So gibt es zahlreiche Einsteiger-Programme, die das nötige Grundwissen vermitteln, ebenso wie Programme für den fortgeschrittenen BASIC-Programmierer. Der Profi hingegen erfährt, wie er die Grafik von C aus ansprechen kann. Kurzum ein Buch, in dem Sie finden, was Sie brauchen: Grafikprogrammierung mit den vorhandenen BASIC-Befehlen, Nutzung der Libraries, die Register der Grafik-Chips, Aufbau und Programmierung von Screens, Windows, HAM, Halfbrite und Interlace aus BASIC und C, 1024 x 1024-Superbitmap, gepufferte Multitasking-Hardcopy-Routine – zum Thema Grafik werden Sie in Amiga Supergrafik nichts vermissen.
Amiga Supergrafik
686 Seiten, DM 59,–

Der Amiga ist das ideale Werkzeug für Kreative – aber einen Film mit ihm erstellen? Warum nicht? Dieses Buch hilft Ihnen dabei. Vom Drehbuch bis zum fertigen Drei-Minuten-Film. Mit allem, was dazu gehört: Grafik, Sound, Animation und und und. Dabei arbeiten Sie mit den verschiedenen DeLuxe-Programmen – auch mit DeLuxe-Video 1.2 und DeLuxe-Paint II, lernen deren fantastische Möglichkeiten und Anwendungsgebiete kennen. Da Sie als angehender Filmemacher auch mit Videorecorder und Sounddigitizer professionell umgehen müssen, wird das nötige Know-how gleich mitgeliefert. Wann dürfen wir zu Ihrem ersten Oskar gratulieren?
Amiga – Der Film
Hardcover, ca. 400 Seiten
DM 49,–
erscheint ca.
Ende 11/87

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
zzgl. DM 5,– Versandkosten
unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort

Diskettenkapazität zu gering?

Dann sollte an ein Speichermedium mit mehr Kapazität gedacht werden. Aus diesem Grund stellen wir Ihnen verschiedene Festplatten vor, die sich zum Anschluß an den Amiga eignen.

Im Laufe der Zeit wird für den einen oder anderen Amiga-Besitzer sicher die Anschaffung eines Festplatten-Laufwerks nötig. Immer umfangreichere Anwendungen und Programme kommen ohne Festplatten schon fast nicht mehr aus (beispielsweise Videoscape 3D oder Datenbanksysteme). Da in den letzten Monaten ein mehr oder weniger großer Preisverfall bei Plattenspeichern festzustellen war, werden Hard-Disks mit mittelgroßen und großen Kapazitäten auch für Käufer mit schmäleren Geldbörsen interessant. Im folgenden stellen wir Ihnen vier verschiedene Festplattensysteme vor, die Ihren Amiga hilfreich aufwerten.

Für schmale Geldbeutel

Amiga 1000-Besitzer, die sich für ein preiswertes Plattenlaufwerk interessieren, werden an der Hard-Disk von Xebec (Bild 1), die von Atlantis vertrieben wird, nicht vorbeikommen. Das Plattenlaufwerk ist übrigens auch für den A500 verwendbar, wenn ein Adapterkabel angeschlossen wird, das den um 180 Grad gedrehten Bus berücksichtigt.

Das Gerät, das mit einem Interface an den Expansion-Bus des A1000 angesteckt wird, ähnelt entfernt an einen kleinen Heizlüfter. Nach Angaben des Herstellers ist aber ein neues Gehäuse in Arbeit, das besser aussehen soll und über Platz für eine zweite Platte oder einen Streamer verfügt. Das 5¼-Zoll-Laufwerk (75 Millisekunden) besitzt eine formatierte Speicherkapazität von 20 MByte und besitzt eine Autopark-Schaltung. Das heißt, daß die Schreib-/Leseköpfe beim Ausschalten des Geräts automatisch arretiert werden (wichtig für den Transport).

Zum Preis von 1248 Mark erhält der Käufer neben dem eigentlichen Gerät mit dem externen Controller eine Installationsdiskette und ein dünnes, deutsches Handbuch. Probleme traten während des Testbetriebs nicht auf, die Platte arbeitete (insofern sie mit der entsprechenden Software aktiviert wurde) anstandslos mit dem Amiga zusammen.

Obwohl die Zugriffsgeschwindigkeiten nicht allzu hoch und die Kapazität mit 20 MByte für Amiga-Verhältnisse fast schon zu wenig ist, kann die Platte aber guten Gewissens jedem A500/A1000-Benutzer mit schmalem Geldbeutel empfohlen werden. Anwender, die schnellere Platten und mehr Kapazitäten benötigen, sollten sich aber besser den Kauf einer teueren SCSI-Platte, wie sie beispielsweise von Intelligent Memory oder DTM angeboten werden, überlegen oder auf die von Xebec geplanten größeren Lauferke mit 40 MByte (2198 Mark) und 70 MByte (3598 Mark) warten.

Die weiteren drei Platten, die wir Ihnen vorstellen, sind derzeit nur für den A2000 geeignet. Sollten aber in Zukunft externe Controller für den A500/A1000 auf dem Markt erscheinen, kann sicher die eine oder andere Platte daran angeschlossen werden.

Erster Leckerbissen ist der von Commodore produzierte und im Fachhandel erhältliche SCSI-Controller »2090« mit der zugehörigen Festplatte »2092« (Bild 2). Die 3½-Zoll-Festplatte (89 Millisekunden) besitzt eine formatierte Kapazität von 20 MByte, die auch voll von Amiga-DOS benutzt werden können. Der Einbau erfolgt mit dem mitgelieferten Montagematerial anstelle des zweiten internen Diskettenlaufwerks.

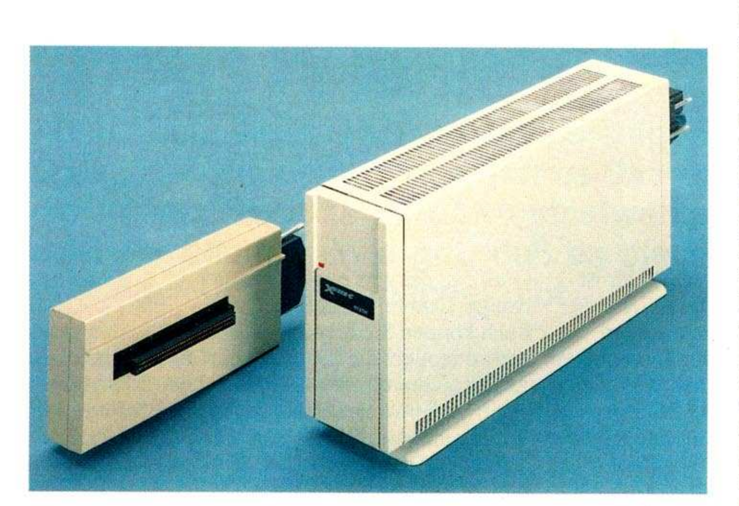


Bild 1. Die 20-MByte-Platte von Xebec zum Anschluß an den A1000 (für den A500 ist ein Adapterkabel nötig)

Die mitgelieferte Software bereitet die Platte zur Formatierung vor und installiert alle zum Betrieb notwendigen Treiberprogramme auf der Startdiskette. Ausgeliefert wird eine 3½-Zoll-Hard-Disk vom Typ »Epson HMD-720«, die nicht über das bereits erwähnte »Autopark« verfügt.

Das wichtigste Teil aber bei dem Festplatten-Set ist der SCSI-Controller. Erst er ermöglicht überhaupt die Benutzung einer (beliebigen) Festplatte. Im Klartext heißt das, daß der Käufer dieser Karte nicht nur auf die mitgelieferte Platte angewiesen ist, sondern jede Hard-Disk, die er möchte, anschließen kann. Der Controller, der in einen der Amiga-Slots gesteckt wird, stellt Anschlußmöglichkeiten für insgesamt neun verschiedene Platten zur Verfügung, von denen einer von der mitgelieferten Platte belegt wird: zwei ST 506-Laufwerke (typische PC-Festplatten) und maximal sieben SCSI-Hard-Disks.

Über den SCSI-Controller können sowohl die PC-/AT-Karte wie auch der Amiga problemlos auf die Hard-Disk zugreifen. Dabei ist das Besondere, daß von nun an von MS-DOS aus (nur mit den neuesten Versionen der bei der PC-Karte mitgelieferten Disketten) keine festen Partitions (Reservierungen von Plattenbereichen) mehr vorgenommen werden müssen. Der MS-DOS-Bereich paßt sich dynamisch den Anforderungen an. Dies muß leider mit dem Preis bezahlt werden, daß in diesem Fall nicht mehr vom PC-Laufwerk »C:« gebootet werden kann (dafür wäre nach wie vor ein PC-Controller nötig). Beim ersten Kurztest erwies sich, daß die Kombination ohne Probleme zu formatieren war und auch das MS-DOS-System zugreifen konnte.

Mehr Kapazität

Das mitgelieferte Handbuch ist leider nicht optimal geschrieben, gibt aber genügend Hilfestellung beim Einbau des Kits. Alles in allem erhält der Käufer für 1695 Mark einen Festplattenkit mit Controller, dessen Flexibilität beim Anschluß von verschiedenen Festplatten keine Probleme für zukünftige Anschaffungen von Zusatzplatten bereitet.

Besitzer eines A2000 mit PC-/AT-Karte benötigen aber nicht unbedingt das Commodore-SCSI-Set. Eine Alternative ist eine Filecard, die in einen freien PC-Slot gesteckt wird (Filecard = Laufwerk und Controller auf einer Karte zum Ein-

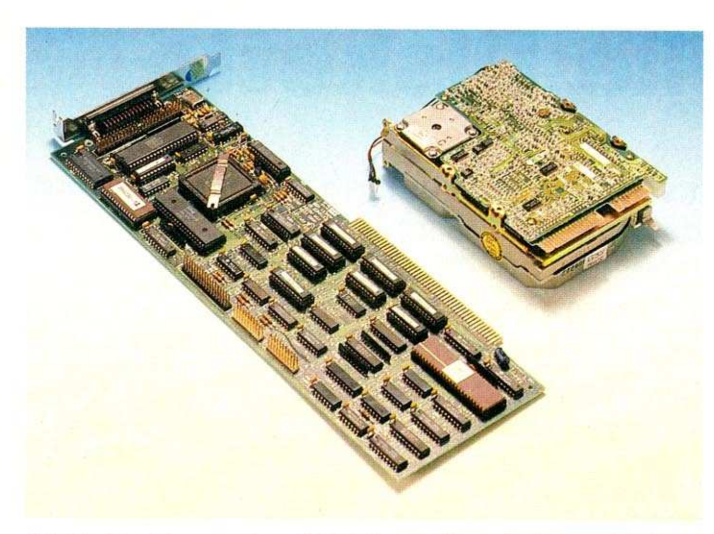


Bild 2. Der Commodore-SCSI-Controller mit dazugehöriger Festplatte zum Einbau in einen Amiga 2000

stecken). Das Praktische daran ist, daß die Platte im Gehäuse des Computers »verschwindet«. Eine Karte, die auf der PC-Seite installiert wird, kann mit den geeigneten Programmen (auf jeder neueren Workbench enthalten) sowohl vom PC-System als auch vom Amiga aus genutzt werden. Vom Amiga aus wird sie mit »jh0« angesprochen. Allerdings müssen bereits beim Formatieren jedem System feste Bereiche der Platte (Partitions) zugeteilt werden (etwa 10 MByte für den PC und 10 MByte für den Amiga). Eine spätere Korrektur der Partitions ist ohne Neuformatieren nicht mehr möglich. Als Beispiel für eine geeignete Filecard dient die »Filecard 40« von CTT (Bild 3 rechts), die bei einem Preis von unter 2000 Mark eine formatierte Kapazität von 40 MByte und Autopark bietet.

Die Filecard belegt zwei lange PC-Steckplätze. Eine kurze Karte findet aber noch Platz im danebenliegenden Slot. Der Einbau der Platte ist kein Problem. Die Installationssoftware, die man auf neueren Workbench- und Bridgeboard-Disketten findet (DJMOUNT, DPFORMAT, ADISK, FDISK), erlaubt ein schnelles »Vorbereiten« der Hard-Disk. Die Filecard, die vom PC-/AT-System automatisch erkannt wird, gestattet das Booten des Betriebssystems wie gewohnt von Laufwerk »C:«.

Beim Einbau und Test traten während der gesamten Dauer keine Probleme auf, beide Systeme arbeiteten ordnungsgemäß mit der Hard-Disk zusammen. Für Besitzer einer PC-/AT-Karte, denen 20 MByte Speicherkapazität nicht genug sind, ist die 40-MByte-Filecard (35 Millisekunden) von CTT sicher eine sinnvolle Alternative.

Die »Große« bei den Platten

Es ist sicher denkbar, daß bei späteren A2000-Anwendungen (beispielsweise Netzwerke) auch Hard-Disks mit genügend großen Kapazitäten nötig werden. In Anbetracht möglicher späterer Bedürfnisse stellen wir Ihnen eine 5¼-Zoll-Platte mit hoher Kapazität vor, die geeignet ist, umfangreiche Speicherplatz-Ansprüche zu befriedigen. Es ist dies die »Priam V185« von CTT (Bild 3 links), die bei normaler Bauhöhe etwa 75 MByte speichern kann. Mit einer gemessenen Zugriffszeit von 24 ms gehört sie zu den schnellen Platten, wobei natürlich auch ein Autopark vorhanden ist. Das Chassis ist mit dem Einbaurahmen über Gummilager verbunden, wodurch Erschütterungen abgefangen werden sollen. Die Platte kann sowohl an den weiter oben besprochenen 2090-SCSI-Controller von Commodore wie auch an einen PC-/AT-Controller angeschlossen werden. Im letzten Fall ist allerdings das Vorhandensein einer PC/AT-Karte nötig, da die Hard-Disk über den PC-Teil angesprochen wird. Im Test wurde mit beiden Controllertypen gearbeitet (OMTI- und 2090-SCSI-Controller). Probleme traten nicht auf.

Einzig beim Einbau gab es Schwierigkeiten. Da die 5¼-Zoll-Festplatte eine Höhe von etwa 85 mm besitzt, kann sie nicht in den A2000 »verstaut« werden. Sie muß also neben dem Computer stehen und durch genügend lange Kabel mit dem Controller verbunden sein. Da die Platte hermetisch verschlossen ist, sind aber keine Beschädigungen durch Staub oder andere Einflüsse zu befürchten.

Zu einem Preis von unter 2200 Mark erhält man ein Gerät, das durch sehr schnelle Zugriffszeiten und viel Speicherplatz überzeugt. Sie kann sowohl an den 2090- wie auch einen PC-Controller angeschlossen werden.

Festplatten sind durch ihren hohen Speicherplatz und die deutlich höheren Zugriffsgeschwindigkeiten sicher ein Weg, um schneller und komfortabler mit dem Amiga arbeiten zu können. Leider sind die Controller für den Amiga 500/1000 noch immer nicht fertiggestellt, so daß Besitzer dieser Geräte, die sich für eine preiswerte Hard-Disk interessieren, derzeit auf die Xebec-Platte zurückgreifen müssen.

Für den A2000 sieht es da schon besser aus: Mit gebräuchlichen PC-Controllern oder dem neuen SCSI-Controller von Commodore steht der Weg offen, beliebige Hard-Disks anzuschließen. Sollten die beschriebenen Platten nicht das Richtige sein, bleibt die Möglichkeit, gängige PC-Zeitschriften wie etwa »Computer persönlich« oder das »PC Magazin« zu studieren, da die dort getesteten Hard-Disks zum Großteil auch mit dem Amiga 2000 genutzt werden können. (dm)

Atlantis, Dunantstr. 53, 5030 Hürth,
Tel. 02233/31066
CTT, Kreillerstr. 21, 8000 München 80,
Tel. 089/4361001

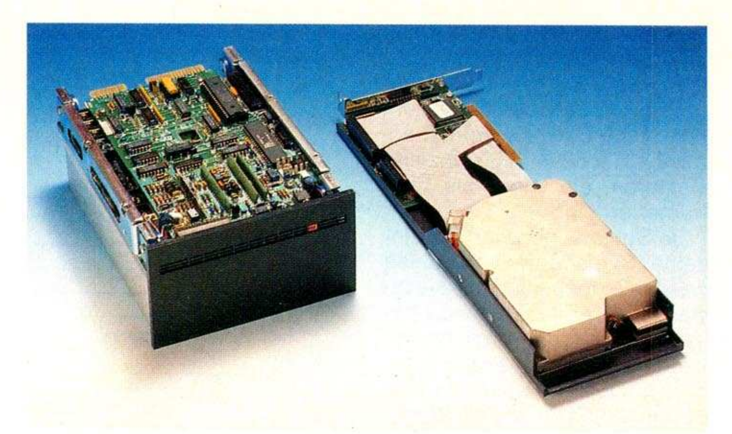


Bild 3. Zwei CTT-Festplatten mit 40 beziehungsweise 75 MByte Kapazität, die sich für den Amiga 2000 eignen

Platte/Anbieter	Kapazität	Geeignet für	Autopark	Größe der Platte	Preis	Bemerkung
Xebec/Atlantis	20 MByte	A500/1000	k.A.	5¼ Zoll	1248 Mark	kein durchgeschleifter Bus; Anschluß an A500 nur mit Adapterkabel
2090/2092-Kit Commodore	20 MByte	A2000	nein	3½ Zoll	1695 Mark	ST 506-Platte, SCSI-Controller für insgesamt neun Plattenlaufwerke
Filecard 40 CTT	40 MByte	A2000 mit PC/AT-Karte	ja	3½ Zoll	unter 2000 Mark	Filecard zum Einstecken in PC/AT-Slot, belegt zwei lange Steckplätze
PRIAM V185 CTT	75 MByte	A2000 mit 2090- oder PC/AT-Controller	ja	5¼ Zoll, volle Bauhöhe	unter 2200 Mark	ohne Controller, zum Einbau in A2000 zu groß, sehr schnelle Platte

Tabelle. Die wichtigsten Daten der besprochenen Festplatten auf einen Blick

Geheimnisvolle Diskette

Kennen Sie sich auf Ihren Disketten aus? Wir sagen Ihnen, wie mit Amiga-DOS formatierte Disketten aufgebaut sind, damit Sie beim Arbeiten mit einem Diskettenmonitor nicht mehr vor »unüberwindlichen Barrieren« stehen.

Wenn direkt auf der Diskette gearbeitet werden soll (etwa mit einem Diskettenmonitor), ist es unabdingbar, näheres über den Aufbau und die Funktion der einzelnen Bits auf Diskette zu wissen. Um Ihnen in diesem Punkt Hilfestellung zu geben, haben wir einmal eine typische Amiga-Diskette »durchleuchtet«. Im folgenden Artikel finden Sie Informationen darüber, wozu die einzelnen Linkpointer da sind und wie man mit ihnen am besten umgeht.

Doch zuerst etwas über den allgemeinen Aufbau einer Diskette: Der Datenträger besteht aus einer flexiblen, runden Scheibe. Auf beiden Seiten der Scheibe ist eine magnetisierbare Schicht aufgebracht. Je nachdem, wie die einzelnen Metallpartikel magnetisiert werden, kann der Schreib-/Lesekopf daraus eine »0« oder eine »1« erkennen. Der Amiga formatiert die Disketten beidseitig mit 80 Spuren (Tracks). Die Tracks sind von 0 bis 79 durchnumeriert. Jeder Track ist in elf Sektoren unterteilt, die von 0 bis 10 numeriert sind und jeweils 512 Byte speichern können. Eine Diskette enthält somit 2 x 11 x 80 = 1760 einzelne Sektoren, die von 0 bis 1759 numeriert sind.

Je Sektor sind zwar 512 einzelne Bytes enthalten, diese werden jedoch, insofern sie etwas mit der Filestruktur zu tun haben, zu Longwords (Longword = vier Byte) zusammengefaßt. Deshalb sieht ein Diagramm eines einzelnen Blocks wie in Bild 1 gezeichnet aus. Der Übersichtlichkeit halber haben wir jeden Block in 32 Zeilen zu je vier Longwords eingeteilt. Die Zahlen am Anfang jeder Zeile geben die hexadezimale Adresse des Longwords im Datenblock an.

An erster Stelle

Unter Amiga-DOS existieren sechs verschiedene Typen von Blöcken, die jeweils eine bestimmte Aufgabe haben. Wir erläutern jeden einzelnen dieser Blocks mit der Funktion, die er innehat, und welchen Zweck die einzelnen Bytes/Longwords haben.

Beginnen wir mit dem wichtigsten Block, der übrigens als einziger Block immer an einer bestimmten Stelle auf der Diskette steht. Es ist dies der ROOT-Block, den Sie auf jeder Diskette im Block 880 ($370) finden. Von diesem Block aus können über die richtigen Pointer (Zeiger) die einzelnen, zum Directory gehörenden Einträge gefunden werden. Betrachten Sie dazu auch Bild 2. Sie sehen, daß vom ROOT-Block aus Pfade zu den verschiedenen anderen Blöcken führen.

Zur Erklärung der einzelnen, für diesen Block wichtigen Longword-Pointern ziehen Sie bitte Bild 3 zu Rate. Sie sehen den bereits in Bild 3 erwähnten Aufbau der Struktur.

Das erste (Adresse $000) und das letzte Longword (Adresse $1FC) sind eine Kennung für den ROOT-Block. Steht in diesen beiden Longwords $00000002 und $00000001, dann handelt es sich um den Hauptblock. Die Felder 2 und 3 haben keine Bedeutung, während das vierte Longword angibt, wie viele Einträge das Linkfeld (Hash-Table) enthält. Dazu aber gleich mehr. Das sechste Feld ($014) enthält das Longword für die Checksumme, mit deren Hilfe beispielsweise ein defektes Byte neu berechnet werden kann. Die Checksumme errechnet sich folgendermaßen: Vom Wert $FFFFFFFF werden der Reihe nach die Werte der einzelnen Longwords (außer Feld 6) abgezogen. Der so entstandene Restwert bildet die Checksumme.

Ab Adresse $018 beginnt eine 72 Longwords große Tabelle, in der die Startblöcke der einzelnen Directory-Einträge enthalten sind.

Dazu eine kleine Anmerkung: Wenn Sie sich schon gewundert haben, warum der Amiga beim Anzeigen des Directories so lange braucht, liegt das in der Pointer-Struktur begründet. Zuerst wird aus der Hash-Table die Adresse des nächsten Directory-Blocks gelesen. Anschließend positioniert Amiga-DOS den Schreib-/Lesekopf auf dem entsprechenden Block, wo der eigentliche Dateiname steht. Dieser wird angezeigt und der Kopf fährt zurück zum ROOT-Block, wo die nächste Eintragsadresse gelesen wird.

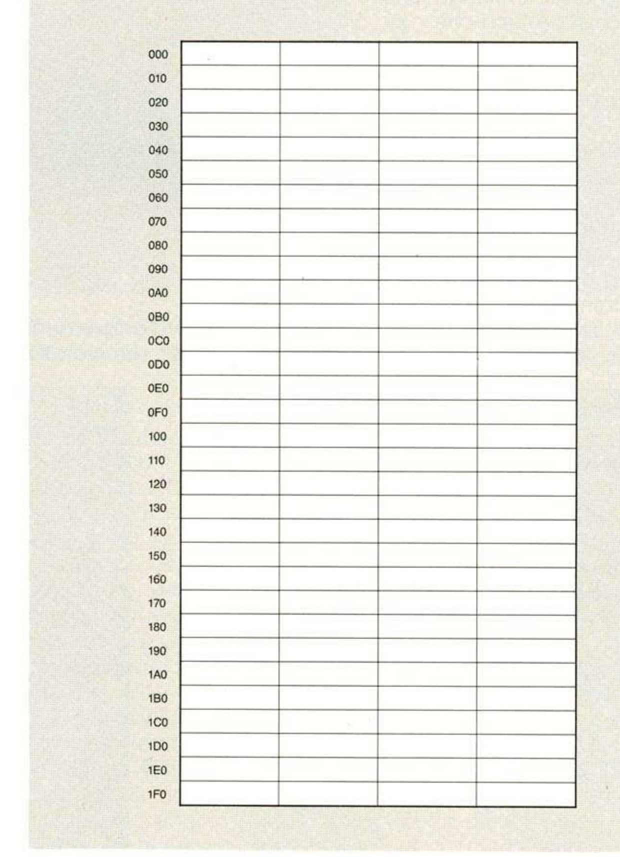


Bild 1. Jeder Block auf der Diskette ist aus insgesamt 128 Longwords oder 512 Byte aufgebaut

ROOT — die Basis

Das folgende Longword ($138) ist eine Gültigkeitsanzeige für die Sektormap. Enthält das Feld den Wert $FFFFFFFF, ist die Sektormap, die die Blockbelegung der Diskette enthält, gültig, die Diskette ist intakt. Andernfalls wird ein Fenster ausgegeben, das die schöne Meldung »Error Validating Disk, use Diskdoctor...« enthält.

Das Feld mit der Adresse $13C stellt den Zeiger auf die Sektormap dar. Dieser Block hat keine feste Blockadresse, er kann irgendwo auf der Diskette zu finden sein. Auf einer frisch formatierten, noch nicht bespielten Diskette liegt er normalerweise im Block 881 ($371). Der Bereich von $140 bis $1A3 ist beim ROOT-Block nicht belegt.

Die folgenden drei Longwords ($1A4, $1A8, $1AC) beinhalten das Datum und die Uhrzeit der letzten Änderung des Blocks. Richtiger heißt es aber, die Zeit der letzten Änderung irgendeiner Datei auf der Diskette. Wie Sie sicher schon gehört haben, nimmt der Amiga beim Start, falls die Uhr nicht mit DATE oder SETCLOCK gestellt wird, das Datum des letzten Diskettenzugriffs als Datumsbasis. Dies garantiert, daß auch bei nichtgestellter Uhr kein späterer Disketteneintrag älteren Datums ist.

Die Uhrzeit ist in diesen drei Longwords nicht im Klartext, sondern in einer zwar kurzen,

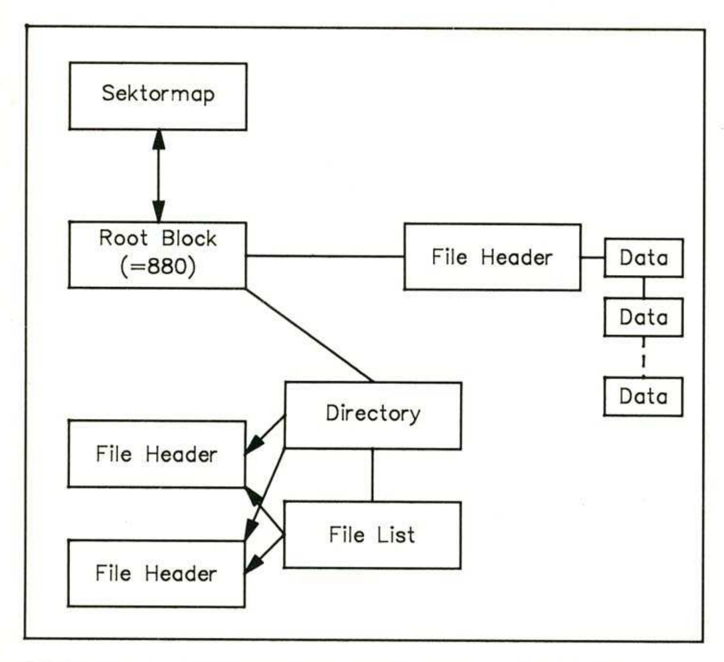


Bild 2. Vom ROOT-Block aus zeigen Pointer auf die einzelnen Directory-Einträge und die Sektormap der Diskette

aber umständlichen Schreibweise abgelegt.

Das Tagefeld beinhaltet die Zahl der einzelnen Tage seit dem 01.01.1978, die Stunden und Minuten werden komplett in Minuten gespeichert. Ein dort enthaltener Wert von $00000132 würde also bedeuten, daß der Eintrag 306 Minuten alt ist. Die Uhrzeit, die ja mit LIST angezeigt werden kann, hieße also 05:06:xx.

Organisation ist alles

Bei den Sekunden wird's noch komplizierter: Der im Sekundenfeld enthaltene Wert muß durch 50 geteilt werden. Ein Eintrag von $00000549 (dezimal 1353) bedeutet 1353/50=27 Sekunden.

Im Anschluß daran (ab $1B0) steht der Diskettenname (sieben Longwords). Der eigentliche Name beginnt erst bei $1B1, da das erste Byte ($1B0) die Zahl der gesamten Buchstaben angibt, aus denen der Name besteht.

Das zweite, drei Longwords umfassende Datumsfeld gibt das Datum der Erstellung der Diskette, also deren Formatierung, an.

Sie sehen also, die verschiedenen Pointer innerhalb eines Blocks haben zum Teil sehr wichtige Funktionen, die der Organisation der Diskettenstruktur dienen.

Der zweite Blocktyp ist die bereits erwähnte Sektormap (Bild 4). Dieser Block enthält die Verfügbarkeitsliste aller 1760 Blocks und besitzt keine Typ-Kennung. Das bedeutet, daß das Betriebssystem der Sektormap entnehmen kann, welche Blocks zur Datenspeicherung frei sind und welche nicht beschrieben werden dürfen (ähnlich der BAM, wie sie vom C 64 und seiner Floppystation her bekannt ist). In diesem Block sind nur die Felder $000 bis $0DC belegt, der Rest wird nicht benötigt. Das erste Longword enthält die erwähnte Checksumme, während die restlichen Longwords für die Belegung der Blocks stehen. In Bild 4 sehen Sie genau, welches Feld für welche Blocks zuständig ist. Bild 5 zeigt exemplarisch, wie ein freier (0) und ein belegter (1) Block gekennzeichnet ist und wie sich das Longword errechnet.

Jede Datei, die im Directory eingetragen ist, besitzt einen speziellen Block (den FILE HEADER), der die für das Betriebssystem wichtigen Daten zum jeweiligen File enthält (Bild 6). Dort ist der Name der Datei, die Länge und das Erstellungsdatum enthalten. Wie auch der ROOT-Block hat dieser Block eine Typ-Kennung (erstes und letztes Longword, Wert $00000002 und $FFFFFFFD) und drei Datumsfelder ($1A4, $1A8, $1AC). Auch die Checksumme befindet sich im sechsten Feld.

Die 72 Einträge große Hash-Table beginnt bei $018 und endet bei $137. Sie enthält aber nicht die Adressen der Directory-Einträge, sondern der Reihenfolge nach (beginnend bei Feld $134) die Adressen sämtlicher Blocks, die zu der Datei gehören (sollten es mehr als 72 Blocks sein, trägt der Amiga die weiteren Blockadressen in FILE LIST-Blöcken ein. Dazu aber später). Dies hat den Vorteil, daß sofort alle Blockadressen einer Datei zur Verfügung stehen, ohne daß sich das Betriebssystem von Pointer zu Pointer »hangeln« muß. Auch wenn ein Datenblock defekt ist, können so noch die folgenden Datenblöcke rekonstruiert werden.

Die Blocknummer des ersten Datenblocks der Datei ist in Feld $010 enthalten. Diese Blockadresse gibt an, ab wo das eigentliche Programm/die Datei startet.

In Feld 2 steht ein Pointer, der immer auf sich selbst, also den jeweiligen FILE HEADER-Block, deutet. Das dritte Longword gibt die Zahl der Blocks an, die in der Hash-Table enthalten sind.

Feld $140 ist für die PROTECT-Flags zuständig. Steht in den letzten vier Stellen eine Null, kann die Datei gelesen, beschrieben, gelöscht und editiert werden. Würde das Feld $00001111 enthalten, wäre die

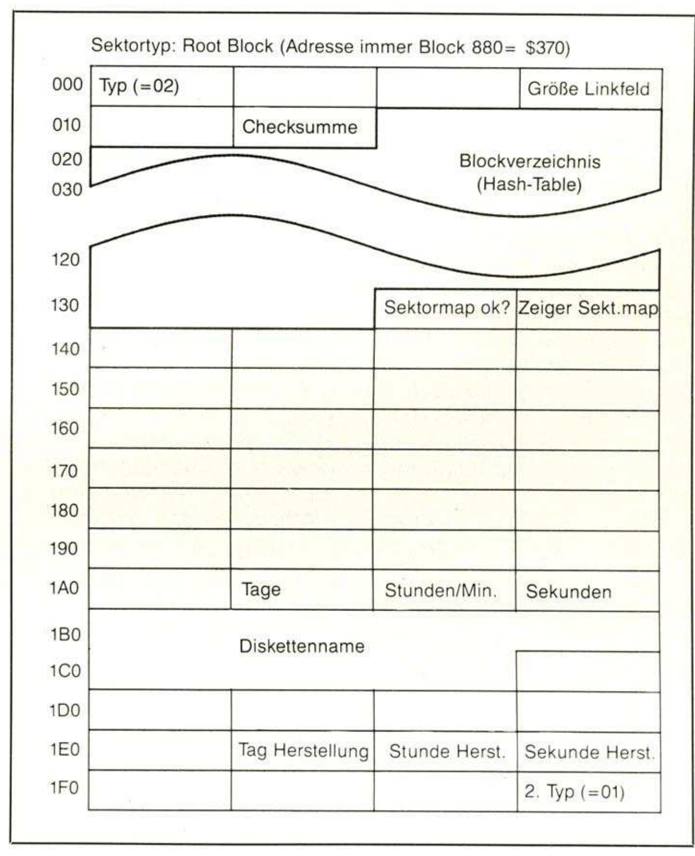


Bild 3. Der Aufbau des ROOT-Blocks (Erklärung siehe Text)

Sektortyp: Sektormap

000	Checksumme	2-33	34-65	66-97
010	98-129	130-161	162-193	194-225
020	226-257	258-289	290-321	322-353
030	354-385	386-417	418-449	450-481
040	482-513	514-545	546-577	578-609
050	610-641	642-673	674-705	706-737
060	738-769	770-801	802-833	834-865
070	866-897	898-929	930-961	962-993
080	994-1025	1026-1057	1058-1089	1090-1121
090	1122-1153	1154-1185	1186-1217	1218-1249
0A0	1250-1281	1282-1313	1314-1345	1346-1377
0B0	1378-1409	1410-1441	1442-1473	1474-1505
0C0	1506-1537	1538-1569	1570-1601	1602-1633
0D0	1634-1665	1666-1697	1698-1729	1730-1759 / 0-1

Bild 4. Die Sektormap informiert, welche Blocks auf der Diskette belegt oder frei sind

Datei gegen diese vier Bearbeitungsmöglichkeiten geschützt. Dieses Longword wird vom CLI aus mit dem PROTECT-Befehl beeinflußt.

Das darauf folgende Longword gibt die gesamte Länge der Datei in Byte an, wobei der FILE HEADER nicht dazu zählt.

Wie bekannt, kann vom CLI aus mit FILENOTE oder auf der Workbench-Ebene mit INFO ein Kommentar zu einer Datei erstellt werden. Der Text dazu steht in den Feldern $148 bis $19C. Der Kommentar wird übrigens beim Kopieren nicht mitübertragen und auch nicht überschrieben, was wohl auf einen Fehler im Amiga-DOS schließen läßt.

Wie bereits erwähnt, kann es vorkommen, daß eine Datei mehr als 72 Blocks belegt. In diesem Fall würde die Hash-Table voll sein. Es muß also auf eine weitere Hash-Table ausgewichen werden, wo die Blockliste fortgesetzt wird. Dazu dient das vorletzte Longword. Es enthält die Adresse des Blocks, bei dem die Hash-Table fortgesetzt wird. Das Feld davor zeigt, insofern es der erste FILE HEADER-Block ist, auf den ROOT-Block, ansonsten auf seinen Vorgängerblock.

Wie ein FILE LIST-Block, also ein Folgeblock für die Hash-Table, aussieht, zeigt Bild 7. Seine Typ-Kennung ist $00000010 und $FFFFFFFD und auch die Checksumme sitzt in Feld sechs. Er enthält hauptsächlich die Hash-Table (von $018 bis $137), in der die Pointer auf die einzelnen Datenblöcke vom FILE HEADER fortgesetzt werden. Das dritte Feld gibt an, wie viele Einträge die Hash-Table umfaßt (maximal 72). Das zweite Feld deutet auf den FILE HEADER, von dem es abstammt. Sollte ein weiterer Block nötig sein, steht dessen Adresse im vorletzten Feld, das Longword davor gibt die Blocknummer des Vorgängerblocks an.

Es geht tiefer

Wie Sie sicher wissen, können sich im Hauptdirectory sowohl Dateieinträge wie auch weitere Directories, sogenannte Subdirectories oder Unterverzeichnisse befinden, die auch wieder Subdirectories beinhalten dürfen. Auf der Diskette wird ein solches Unterverzeichnis durch einen DIRECTORY-Block dargestellt (Bild 8). Er ist prinzipiell wie der ROOT-Block aufgebaut, wobei er keine eigene Sektormap besitzt. Seine Typ-Kennung ist $00000002 und $00000002. Die Checksumme findet sich im sechsten Feld, ein Zeiger auf sich selbst steht im zweiten Longword. Das Datumsfeld wurde bereits weiter oben beim ROOT-Block beschrieben, Informationen zum PROTECT-Feld ($140) finden sich beim FILE HEADER.

Auch zu Unterverzeichnissen können mit dem CLI-Befehl FILENOTE oder von der Workbench-Ebene aus Kommentare eingegeben werden. Der Text dazu findet sich in den Feldern $148 bis $19C. Wie bereits erwähnt, wird er nicht mitkopiert und auch nicht gelöscht.

Die einzelnen Subdirectory-Einträge finden sich in der Hash-Table, die von Feld $018 bis $134 reicht. Sollte sie voll sein (maximal 72 Einträge), wird ein Folgeblock eingerichtet, der die weiteren Blockadressen enthält. Die Blockadresse des Folgeblocks wird durch Longword $1F0 bestimmt. Der Wert des Longwords ist $00000000, wenn kein Folgeblock besteht. Ebenso gibt es ein Feld, das die Blockadresse des Vorgängerblocks enthält. Der Vorgängerblock ist in diesem Fall das jeweils übergeordnete Directory, in welcher das Subdirectory »eingebettet« ist.

Wozu ist eine Diskette eigentlich da? Natürlich, um darauf Daten zu speichern. Bis

Block	Belegt (1 = ja)	Wert
1122	1	
1123	1	
1124	0	D
1125	1	
1126	1	
1127	1	
1128	0	D
1129	1	
1130	1	
1131	1	
1132	1	F
1133	1	
1134	0	
1135	0	
1136	0	1
1137	1	
1138	1	
1139	1	
1140	1	F
1141	1	
1142	0	
1143	1	
1144	1	6
1145	0	
1146	1	
1147	0	
1148	1	B
1149	1	
1150	1	
1151	1	E
1152	1	
1153	0	
LongWord = DDF1F6BE		

Bild 5. Ein Longword, das belegte Blöcke angibt

Sektortyp: File Header
000 Typ (=02) | Zeiger auf sich selbst | Zahl der Blocks
010 1. Dateiblock | Checksumme
020
110 Blockverzeichnis (Hash-Table)
120
130
140 Protect-Flags | Länge der Datei
150
160
170 Kommentar
180
190
1A0 Tage | Stunden/Min. | Sekunden
1B0 Dateiname
1C0
1D0
1E0
1F0 Vorgängerblock | Folgeblock | $FFFFFFFD

Bild 6. Ohne den FILE HEADER könnte das Betriebssystem keine Datei finden

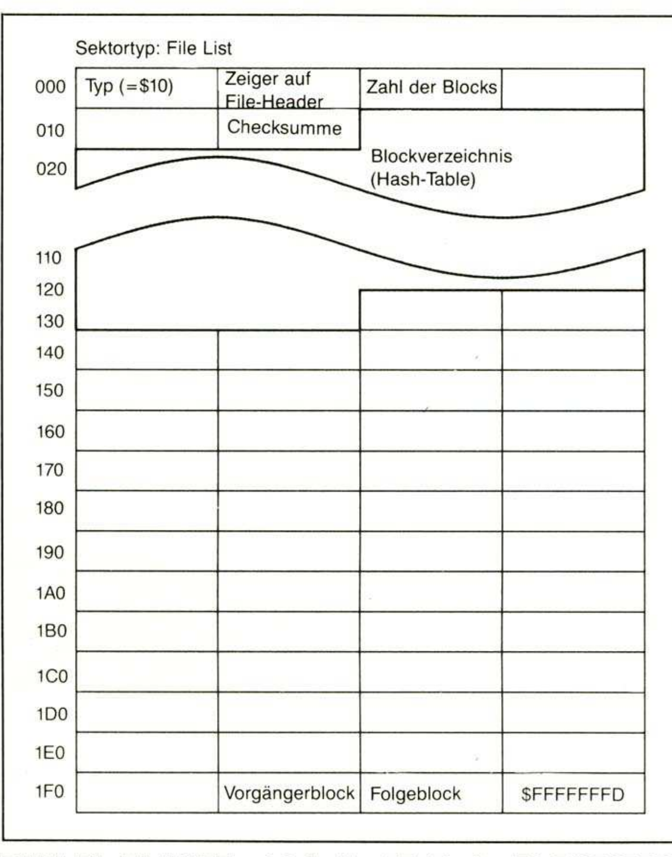


Bild 7. Die FILE LIST setzt die Hash-Table des FILE HEADER-Blocks fort

Amiga 2000
4 MB Steckkarte
2 MB bestückt
999,–

Kupke Computertechnik GmbH

6 Gründe dafür Golem

Drives & Ram Box

3,5 Zoll externe Amiga-Laufwerke
Amiga 500/1000/2000

auch als Einschubdrive für Amiga 2000 **299,–**

1. 100% kompatibeles NEC 1036 a
2. Amiga-farbenes Metallgehäuse
3. farblich passende Frontblende
4. durchgeführter Floppybus
5. Ein-/Aus-Schalter
6. Side-car und PC-Karten kompatibel

369,–



externe RAM-BOX Amiga 1000
auch für Amiga 500 (mit Adapter)

Golem 2 MB erweitert den „Amiga“ auf 2,5 MB

1. Autokonfigurierend (ab Kick 1.2)
2. Amiga-farbenes Metallgehäuse
3. durchgeführter Systembus
4. Ein-/Aus-Schalter
5. Erweiterbar
6. Preis

2 MB **998,–**

5,25 Zoll Amiga Laufwerk
40/80 Track Umschaltung, Ein-/Aus-Schalter, helle Frontblende, Busdurchführung, 880 KB Speicherkapazität, Side-car und PC-Karten kompatibel **479,–**

500er Peripherieadapter
für 1000er Peripherie am 500er **59,90**

Golem Sound
Ein Audiodigitizer der Extraklasse, kompatibel zu „Perfect Sound“ u. „Future Sound“ Software. DIN- und Cincheingang, Empfindlichkeitsregelung für Mikrophonanschluß **138,–**

Wir liefern im 3-Tage-Rhythmus

Kupke Computertechnik GmbH
4600 Dortmund, Apelank 28
Tel.: 02 31/85 26 05

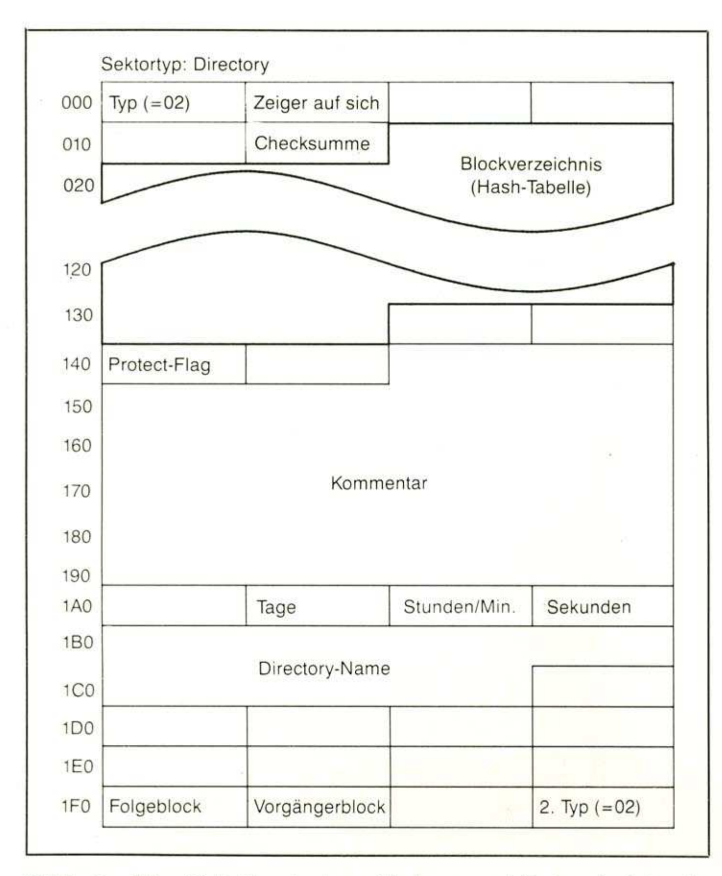


Bild 8. Für Subdirectories (Unterverzeichnisse) ist ein eigener Block, der DIRECTORY-Block zuständig

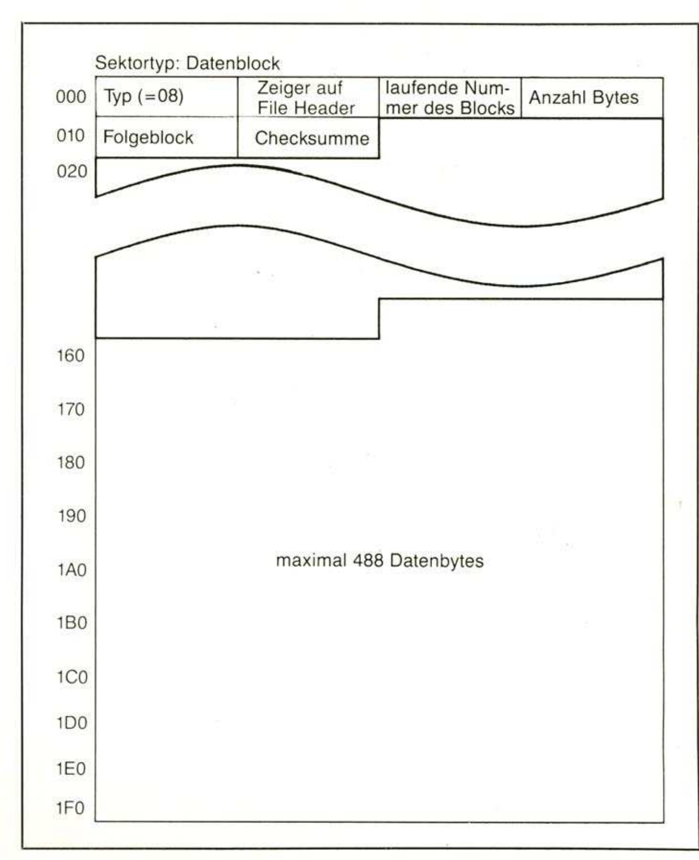


Bild 9. Datenblöcke enthalten insgesamt 488 Datenbytes, die zum eigentlichen Programm gehören

jetzt haben Sie nur Blocktypen kennengelernt, die der Organisation der Diskettenstruktur dienen. Der eigentliche Block, der die einzelnen Bytes der Programme und Dateien enthält, ist der DATENBLOCK (Bild 9). Von den insgesamt 512 Byte, die ein Block fassen kann, dienen sechs der Organisation der Diskettenstruktur, so daß insgesamt 488 Byte je Block für die reine Datenspeicherung verwendet werden können. Der DATA-Block, der durch die Typ-Kennung $00000008 im ersten Longword erkennbar ist, besitzt ebenfalls die schon besprochene Checksumme, die sich im Feld 6 findet. Das zweite Longword ist ein Zeiger auf den FILE HEADER der Datei. Dies hat den Zweck, daß von jedem Datenblock sofort der zugehörige FILE HEADER-Block gefunden werden kann, in dem, wie bereits besprochen, alle wichtigen Daten des Files zu finden sind.

Das dritte Longword gibt an, um den wievielten Block es sich in der Datei handelt.

Um den Folgeblock zu finden, bedient sich das Betriebssystem der Hilfe des fünften Feldes, in dem die Blocknummer des nächsten Datenblocks angegeben ist.

Schließlich und endlich gibt das Longword im vierten Feld noch an, wie viele Byte in dem jeweiligen Block zur Datei gehören (normalerweise 488).

Mit dem Wissen, das Sie in diesem Artikel erhalten haben, sollte es Ihnen nun keine großen Schwierigkeiten mehr bereiten, sich auf einer Diskette mit einem Diskettenmonitor zurechtzufinden. Sei es, um defekte Disketten zu rekonstruieren, Daten zu ändern oder einfach nur des Experimentierens wegen. Wer weiß was alles machbar ist, wenn man sich auf der Diskette auskennt? Doch seien Sie wachsam! Schnell sind mit einem Diskettenmonitor wertvolle Dateien zerstört, die schwer oder gar nicht mehr rekonstruiert werden können.

(dm)



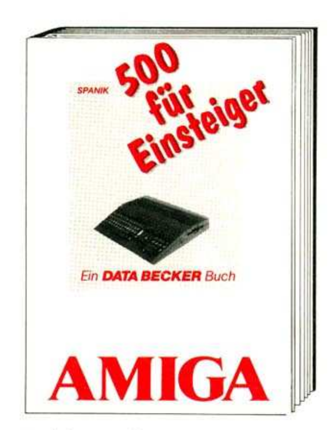




AMIGA BUCHHITS



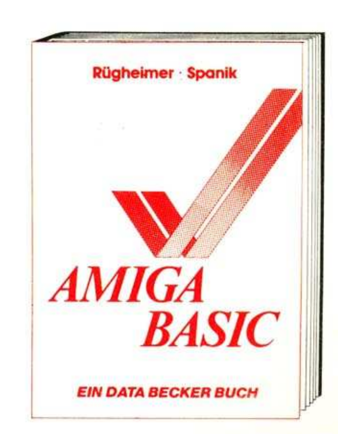
Vom C64 zum Amiga? Nichts leichter als das. Mit dem Aufsteigerbuch. Neben einem schnellen Einstieg in Workbench, CLI und Multitasking findet der Aufsteiger hier einen Einblick in die wichtigsten Sprachen des Amiga – alles immer im Vergleich zu Ihrem alten Rechner. Für den eigentlichen Daten-Transfer von Texten, Bildern, Dateien und BASIC-Programmen enthält das Buch fertige Konvertierungsprogramme. Einen leichteren Umstieg werden Sie nicht finden.
Amiga Aufsteigerbuch.
Hardcover, 320 Seiten, DM 39,–



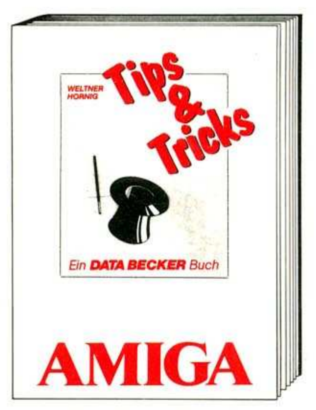
Wählen Sie gleich den richtigen Einstieg zu Ihrem Amiga 500. Denn das Handbuch läßt Sie dabei völlig allein. Versuchen Sie es lieber gleich mit Amiga 500 für Einsteiger. Hier heißt es: anschließen und loslegen. Verständlich für jedermann zeigt Ihnen dieses Buch: Workbench, AmigaBASIC, CLI und AmigaDOS. Locker aufbereitet bietet es Ihnen alles Wissenswerte. Bis hin zu den beim Amiga 500 mitgelieferten Zusatzprogrammen.
Amiga 500 für Einsteiger
343 Seiten, DM 39,–



Läßt das Handbuch Sie auch in so manchen Dingen allein, das große Amiga 2000 Buch nicht. Hier finden Sie eine umfassende Einführung in die Arbeit mit Ihrem neuen Rechner – und mehr als das. Sind Sie erst einmal mit dem Amiga 2000 „per Du", zeigen Ihnen die Autoren, was einen Amiga-Profi ausmacht: Kickstart im RAM, PC-Audioausgänge, erste Hilfe bei Harddisk-Abstürzen, Laufwerkeinbau in den Amiga 2000 und, und, und. Sollten Sie also noch Fragen zu Ihrem Rechner haben, hier finden Sie die Antworten.
Das große Amiga 2000 Buch
Hardcover, 684 Seiten, DM 59,–



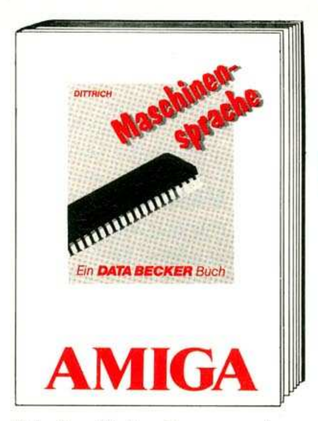
Das erfolgreiche Buch zu Amiga-BASIC – jetzt in Neuauflage! Erweitert um Kickstart 1.2, neuer Workbench und Amiga 500 & 2000. Mit allem, was BASIC-Programmierern Spaß macht: Grafik und Sound, Laden und Speichern von Graficraft-Bildern in BASIC-Programme, sequentielle und relative Dateien, Businеß-Grafik, Computeranimation, Windows, Umgang mit IFF-Bildern, Sprachausgabe und, und, und. Das Buch für Einsteiger, Aufsteiger und Profis.
AmigaBASIC
Hardcover, 774 Seiten, DM 59,–



Amiga Tips & Tricks. Ein Buch, das voller Überraschungen steckt: 64 Farben gleichzeitig auf dem Amiga. Von BASIC aus Zugriff auf die Libraries. Benutzung verschiedener Zeichensätze in BASIC. Sinnvoller Einsatz von Windows, Screens und Menüs. Tips zu einzelnen Grafikbefehlen, Programm- und AmigaDOS-Routinen! Greifen Sie in die Trickkiste, und schon sind Dinge möglich, die man gar nicht gedacht hätte.
Amiga Tips & Tricks
Hardcover, 364 Seiten, DM 49,–

SCHAUN
C für Einsteiger
Ein DATA BECKER Buch
AMIGA

C an einem Wochenende? Durchaus möglich! Mit C für Einsteiger. Ein Einführungskurs, der Ihnen schnell und einfach die wichtigsten Grundlagen dieser Sprache vermittelt. Vom ersten Programm bis hin zu den Routinen in den Bibliotheken. Mit dem gesamten Sprachumfang und den besonderen Features von C. Zahlreiche Tips & Tricks zur Programmierung und eine Beschreibung der beiden Compiler Lattice C und Aztek runden das Ganze ab.
Amiga C für Einsteiger
293 Seiten, DM 39,–



Schreiben Sie Ihre Programme in Maschinensprache – und Sie werden sehen, wie schnell ein Amiga sein kann. Das nötige Know-how liefert Ihnen dieses Buch: Grundlagen des 68000, das Amiga-Betriebssystem, Druckeransteuerung, Diskettenoperationen, Sprachausgabe, Windows, Screens, Register, Pull-Down-Menüs ... Und damit Sie auch gleich praktisch arbeiten können, werden die wichtigsten Assembler vorgestellt.
Amiga Maschinensprache
Hardcover, 288 Seiten
DM 49,–

Super

Regelmäßig in der DATA WELT: Amiga Window – das Forum für den engagierten Amiga-Anwender. Mit kreativen Projekten, Interviews, Software-Tests und wichtigen News. Und: Auch was sich sonst so in der Computerszene tut, erfährt der Amiga-Anwender. Die DATA WELT – das aktuelle Computermagazin. Monat für Monat überall da, wo es Zeitschriften gibt.

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010



Superbase – das relationale Datenbank-System

Superbase vereint als erstes Programm einer neuen Generation von Datenbank-Systemen sowohl eine neuartige, äußerst benutzerfreundliche Bedienung mit Pull-down-Menüs, Fenstern und Maussteuerung als auch die enorme Leistungsfähigkeit einer relationalen Dateiverwaltung.

Einfacher Datenbank-Aufbau

Mit den leichtverständlichen Pull-down-Menüs und Kontrollfeldern legen Sie in Minuten eine komplette Datenbank an. Sie können ein bereits festgelegtes Format jederzeit ändern, ohne Ihre Daten zu zerstören.

Verwaltung der Daten

Superbase zeigt Ihre Daten auf verschiedene Arten an, beispielsweise als Tabelle oder als Formular. Sind Index und Felder selektiert, so können Sie Ihre Daten wie bei einem Videorecorder anzeigen lassen. Schneller Vorlauf, Rücklauf, Pause und Stop – ein Recorder ist nicht einfacher zu bedienen. Ein einzigartiges Filtersystem wählt beliebige Datenkategorien aus, mit denen Sie dann arbeiten können.

Die Stärken von Superbase

Das Festlegen von Übersichten und zusammenhängenden Abfragen über mehrere verknüpfte Dateien ist auch bei verschiedenen Sortierkriterien kein Problem. Daten anderer Datenbanken oder Anwenderprogramme lassen sich ebenfalls problemlos verarbeiten. Binden Sie Daten in Ihre Textverarbeitung ein oder bilden Sie aus verschiedenen Dateien eine neue Datenbank! Die fortschrittliche Baumstruktur und die Disketten-Pufferung garantieren immer höchste Leistungsfähigkeit – Superbase findet beispielsweise einen bestimmten Datensatz in einer Datei, die 100 Adressen umfaßt, in nur 0,5 Sekunden.

Datenbank mit Bildern

Superbase bietet neben den gängigen Datenbank-Funktionen die Möglichkeit, Bilder und Grafiken darzustellen und zu verwalten. Einzigartigen Grafik-Datenbanken oder Dia-Shows steht also nichts im Wege.

Wer braucht Superbase?

Die Anwendungsmöglichkeiten sind nahezu unbegrenzt.

Hier einige Beispiele:

Geschäftliches	Professionelle Anwendungen
Lagerbestand Fakturierung Registratur Versandlisten Verwaltung Adressen	Design Fotografie Journalismus Sammlungen Forschung Ausbildung

* Unverbindliche Preisempfehlung

Leistungsumfang

Die Software: ● bis zu 17 Gigabyte Speicherkapazität pro Datei ● bis zu 16 Millionen Datensätze pro Datei ● maximal 999 Indizes pro Datei ● Anzahl der geöffneten Dateien, Anzahl der Dateien und Anzahl der Felder pro Datensatz: jeweils systemabhängig. Zum Beispiel: Für eine übliche Adreßverwaltung bei einer Datensatzlänge von 200 Byte können Sie auf Ihrer Diskette (880 Kbyte freier Speicher) ca. 4000 Adressen speichern.

Die Daten: ● Text, Daten, numerische Felder und externe Dateien ● Überprüfung bei der Eingabe ● Formelfelder ● Kalender der Jahre 1–9999, verschiedene Datumsformen ● verschiedene Zahlenformate bei 13stelliger Genauigkeit ● Datenschutz per Paßwort

Die Ausgaben: ● das Programm beherrscht einen flexiblen Etikettendruck und produziert übersichtliche Listen mit dem Reportgenerator ● bis zu 255 Spalten ● mit Titel, Datum und Seitenzahl ● Datensatz-Zähler, Durchschnitt, Zwischen- und Endergebnis ● Ausgabe von mehreren Dateien auf Bildschirm, Drucker, Diskette oder neuer Datei ● mehrspaltiger Etikettendruck mit variablem Format ● Speicherung der Ausgabe- und Abfrage-Formate zur späteren Verwendung ● vielfältige Sortierkriterien

Hardware-Anforderung

Amiga mit mindestens 512 Kbyte RAM, beliebiger Drucker mit Centronics-Schnittstelle.

Best.-Nr. 51636
DM 249,–* (sFr 199,–*/öS 2490,–*)

Übrigens: Superbase gibt es auch für Atari ST, Schneider PC und IBM-PCs und Kompatible



710226

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

Zu zweit geht's besser

Ein zweites Diskettenlaufwerk erleichtert das Arbeiten mit dem Amiga ungemein. Wir stellen Ihnen die wichtigsten Zweitlaufwerke in einer Übersicht vor.

Es ist schon ein Graus: Sie arbeiten beispielsweise mit dem CLI (Command Line Interface) und möchten das Directory von einer anderen Diskette anzeigen lassen. Sie wechseln also die Disketten, geben DIR ein und der Amiga verlangt nach der Startdiskette. Also Startdiskette einlegen, DIR ? eintippen, Befehl laden lassen, Diskette wechseln und Return-Taste drücken. Eine umständliche Prozedur, die da erledigt werden muß, um den Disketteninhalt zu sehen. Wieviel leichter ist es, mit zwei (oder mehr) Laufwerken zu arbeiten. Solange sich die Startdiskette in einem Laufwerk befindet, kann die andere nach Herzenslust bearbeitet werden, ohne daß Diskettenwechsel anfallen.

Für alle Amiga-Besitzer, die sich gerne eines dieser hilfreichen Zusatzgeräte zulegen möchten, haben wir eine Übersicht über die auf dem Markt erhältlichen Laufwerke angefertigt. Sie finden in der Tabelle am Ende des Artikels die wichtigsten Daten der handelsüblichen Diskettenstationen zusammengefaßt. »k.A.« bedeutet, daß keine Angaben zum jeweiligen Punkt vorlagen. Die Adressen der einzelnen Hersteller/Anbieter entnehmen Sie bitte dem Schluß der Übersicht.

Vieles wird komfortabler

Bei der Wahl der richtigen Floppystation sind einige Gesichtspunkte zu beachten. Zuerst einmal ist die Frage zu klären, ob ein Zweitlaufwerk genügt oder nicht doch lieber mehrere Drives angeschafft werden sollen. Für den letzten Fall existieren Geräte, die zwei Laufwerke in einem Gehäuse enthalten. Im Regelfall ist dies billiger als der Kauf von zwei Einzellaufwerken. In Bild 1 sehen Sie drei Beispiele: links das Modell von Vesalia Versand, in der Mitte den Flesch & Hörnemann-Drive und rechts das Doppellaufwerk von CompWare. Bei allen diesen Geräten werden die einzelnen Drives als DF1 und DF2 (beziehungsweise DF2 und DF3) angesprochen.

Wichtig ist auch, ob die Drives genügend gegen Fremdeinwirkungen (Magnetfelder oder Ähnliches) abgeschirmt sind und ob die Stabilität des Gehäuses gewährleistet ist. Die Laufwerke sind zwar im Regelfall alle abgeschirmt, doch ist es sicherer, ein stabiles Metallgehäuse als Schutz für den Drive zu wissen. Gute Beispiele dafür sind die Diskettenstationen, die Sie in Bild 2 sehen: Microware (links), C.A.S. (Mitte links), Kupke (Mitte rechts) und Vesalia Versand (rechts).

Besitzen Sie einen Amiga 2000 mit PC/AT-Karte oder einen Sidecar oder wollen Sie mit dem MS-DOS-Emulator 5¼-Zoll-Disketten beschreiben oder lesen? Dann werden Sie wohl nicht umhin kommen, sich ein 5¼-Zoll-Zusatzlaufwerk zu kaufen. Ein Beispiel dafür sehen Sie in Bild 3 links (Vesalia Versand). Oder stehen Sie mehr auf kleine, kurze Gehäuse? Dann gefallen Ihnen sicher die aus Bild 3 ersichtlichen, kompakten Laufwerke von Bittendorf (Mitte links), AB-Computersysteme (Mitte rechts) und noch einmal Bittendorf (rechts). Diese Winzlinge haben den Vorteil, daß sie relativ wenig Platz auf dem Arbeitstisch verbrauchen.

Wenn aber genügend Freiraum auf dem Tisch vorhanden ist, tut's sicher auch ein größeres NEC 1035-Laufwerk. Diese Laufwerke bieten beispielsweise Vesalia Versand (Bild 4 links), ProMicro (Bild 4 Mitte) und Bittendorf an (Bild 4 rechts).

Für gehobene Ansprüche ...

Wer lieber etwas Extravagantes besitzen möchte, sollte das 3½-/5¼-Zoll-Doppellaufwerk von C.A.S. (Bild 5) in Betracht ziehen. Es ist in ein Plexiglasgehäuse eingebaut, damit man die Laufwerke und das

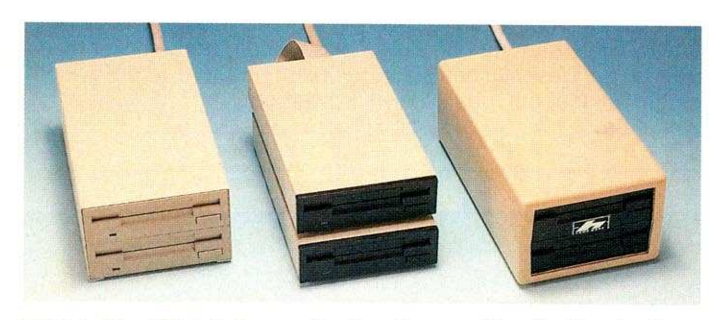


Bild 1. Die 3½-Zoll-Doppellaufwerke von Vesalia Versand, Flesch & Hörnemann und CompWare

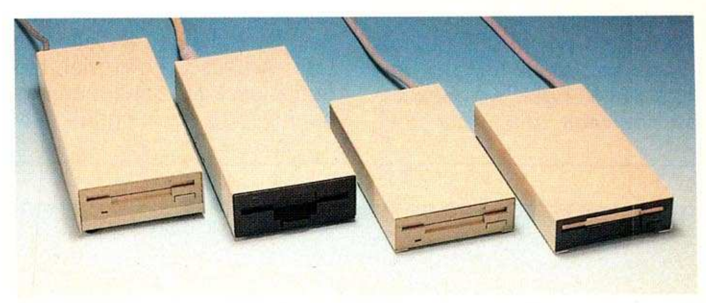


Bild 2. Flaches Design: Zweitlaufwerke von Microware, C.A.S., Kupke Computertechnik und Vesalia Versand

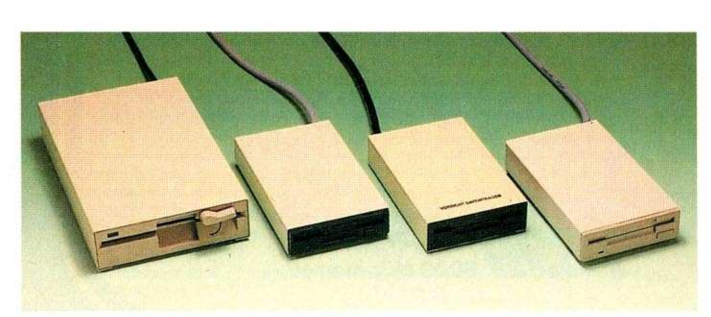


Bild 3. Klein, aber fein: Zweitlaufwerke von Vesalia Versand, Bittendorf, AB-Computer und Bittendorf

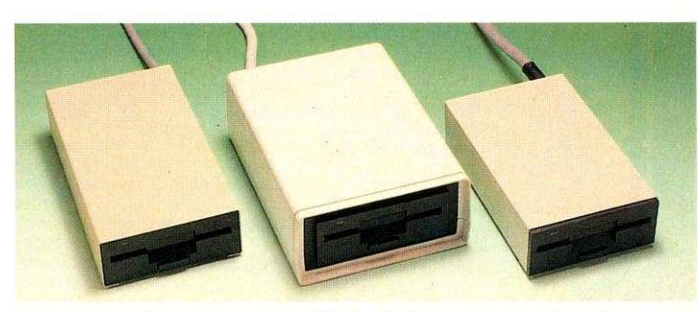


Bild 4. Jeweils mit großen NEC 1035 bestückt: Vesalia Versand, ProMicro und Bittendorf

Drumherum genau betrachten kann.

Was Sie beim Kauf oder der Bestellung aber unbedingt bedenken sollten, ist die Frage, ob zu einem späteren Zeitpunkt weitere Laufwerke gekauft werden könnten. Wenn dies der Fall ist, ist es besser, sich gleich ein (etwas teureres) Gerät zu besorgen, bei dem der Bus durchgeführt ist. Sie erleichtern sich so den Anschluß einer weiteren Diskettenstation. Ein-/Ausschalten der Diskettenstationen sind nicht unbedingt erforderlich, können aber bei Speicherknappheit das Laufwerk abschalten, so daß von AmigaDOS kein Pufferspeicher für angeschlossene Drives verschenkt wird.

Was noch . . ?

Legen Sie Wert auf eine einheitliche, formschöne Computeranlage? Dann sollten Diskettenstationen gekauft werden, die im Farbton zum Amiga-Beige passen. Zur Funktion trägt dies allerdings nichts bei, es ist halt nur was fürs Auge. Egal, welches Zusatzlaufwerk Sie sich (vielleicht zu Weihnachten?) kaufen, eine sinnvolle Aufwertung des Systems ist es allemal. Das Arbeiten geht viel reibungsloser vonstatten, da das nervtötende Einlegen der Startdiskette bei den meisten Dateioperationen entfällt. Und so viel Erleichterung und Zeitgewinn für Wesentliches sollte einem schon ein paar Mark wert sein. (dm)

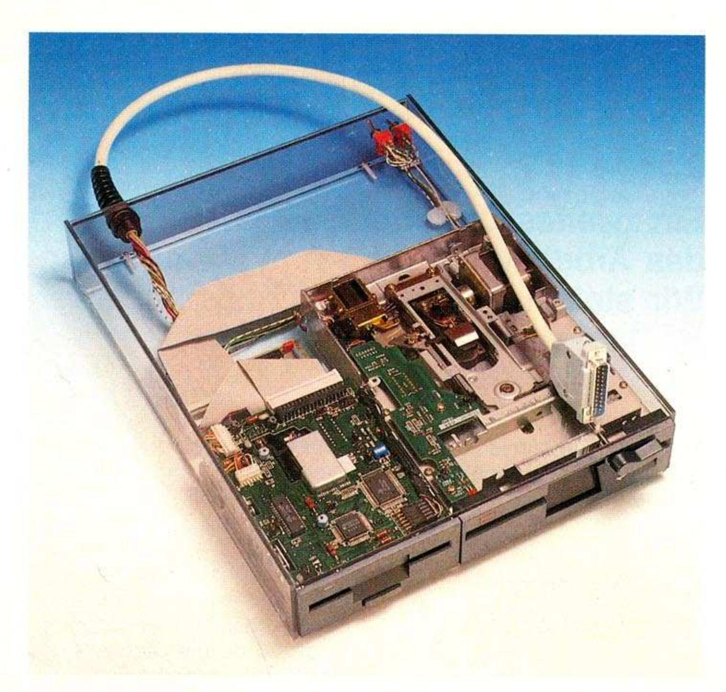


Bild 5. Extravagant: Das Doppellaufwerk von C.A.S. aus Plexiglas zum Beobachten der Laufwerkstätigkeit

In letzter Minute . . .

Kurz vor Redaktionsschluß erreichte uns noch ein weiteres Laufwerk. Das von Bittendorf zu einem Preis von 299 Mark erhältliche neue $5\frac{1}{4}$-Zoll-Laufwerk (NEC oder Fujitsu), das in ein stabiles Metallgehäuse eingebaut ist, bestitzt leider keinen durchgeschleiften Bus (mit Bus 20 Mark Aufpreis), läßt sich aber beispielsweise als Zusatzlaufwerk für MS-DOS oder den Amiga verwenden. Es hat eine 40/80-Track-Umschaltung und ist grau lackiert mit schwarzer Frontblende. Außerdem erkennt es automatisch das Wechseln einer Diskette (Diskchange).

Die Übersicht erhebt keinen Anspruch auf Vollständigkeit. Mögliche Preisänderungen, die nach Redaktionsschluß von den Herstellern/Anbietern bekanntgegeben wurden, konnten nicht mehr berücksichtigt werden.

Hersteller/Anbieter	Gehäuse	Bus durchgeschleift	Ein-/Ausschalter	Laufwerk	3½-/5¼-Zoll	Farbe Gehäuse	Farbe Frontblende	Preis in Mark	Bemerkung
AB Computersysteme	Metall	nein	ja	NEC 1036	3½	Amiga-beige	schwarz	339,—	—
CompWare	Plastik	nein	nein	2 x NEC 1036	2 x 3½	beige	schwarz	650,—	—
ASH Computerware	Metall	nein	nein	NEC 1036	3½	beige	beige	474,—	—
	Metall	nein	ja	Teac	5¼	beige	k.A.	559,—	40/80 Tracks
B. Soltau Electronics	Metall	k.A.	k.A.	NEC	3½	k.A.	k.A.	599,—	—
	Metall	k.A.	k.A.	Teac	5¼	k.A.	k.A.	837,—	—
C.A.S.	Plexiglas	nein	ja	NEC/Teac	3½/5¼	durchsichtig	schwarz	998,—	Doppellaufwerk
	Metall	ja	ja	Teac	5¼	Amiga-beige	k.A.	548,—	—
	Metall	nein	ja	NEC 1036	3½	Amiga-beige	k.A.	529,—	als Doppellaufwerk 849 Mark
	Metall	nein	nein	NEC	3½	k.A	k.A.	398,—	mit Bus 428 Mark
CompTec	k.A.	ja	ja	Teac	2 x 5¼	k.A.	k.A.	920,—	
	Metall	nein	ja	Teac	5¼	k.A.	k.A.	444,—	40/80 Tracks
	k.A.	ja	ja	NEC 1035	2 x 3½	k.A.	k.A.	600,—	Bausatz
	k.A.	ja	ja	NEC 1036	2 x 3½	k.A.	k.A.	655,—	—
	k.A.	ja	ja	NEC 1036	3½	k.A.	k.A.	255,—	Bausatz
	Metall	ja	ja	NEC 1036	3½	Amiga-beige	beige	349,—	mit Bus 366 Mark
Robert Daam Computer	k.A.	ja	ja	NEC 1036	3½	k.A.	k.A.	379,—	—
Datentechnik Bittendorf	Metall	nein	ja	NEC 1035	3½	beige	schwarz	279,—	—
	Metall	nein	ja	Fujitsu	3½	Amiga-beige	schwarz	285,—	33 mm hoch
	Metall	nein	ja	NEC 1036	3½	Amiga-beige	beige	279,—	—
	Metall	nein	ja	Fujitsu	5¼	Amiga-beige	grau	299,—	40/80 Tracks
Dolphin Software	Metall	k.A.	k.A.	NEC 1036	3½	k.A.	k.A.	379,—	—
Flesch & Hörnemann	Metall	nein	nein	NEC 1036	2 x 3½	beige	schwarz	748,—	—
	Metall	ja	ja	NEC 1036	3½	beige	schwarz	398,—	—
	Metall	ja	ja	NEC 1036	3 x 3½	beige	schwarz	1098,—	3 Laufwerke
	Metall	ja	ja	NEC 1036	2 x 3½	beige	schwarz	748,—	—
	Metall	ja	ja	Teac	5¼	beige	schwarz	449,—	40/80 Tracks
	Metall	ja	ja	NEC/Teac	3½/5¼	beige	schwarz	848,—	40/80 Tracks
	Metall	ja	nein	NEC 1036	3½	beige	schwarz	369,—	—
Futurevision	Metall	nein	ja	NEC 1036	3½	Amiga-beige	beige	329,—	—
	Metall	nein	ja	Teac	5¼	Amiga-beige	beige	449,—	—
Interplan	k.A.	k.A.	k.A.	NEC	3½	k.A.	k.A.	568,—	—
	k.A.	k.A.	k.A.	NEC	2 x 3½	k.A.	k.A.	920,—	—
	k.A.	k.A.	k.A.	NEC	2 x 3½	k.A.	k.A.	920,—	—
	k.A.	k.A.	k.A.	Teac	5¼	k.A.	k.A.	568,—	—
Kupke Computertechnik	Metall	ja	ja	NEC 1036	3½	Amiga-beige	beige	369,—	
	Metall	ja	ja	Teac	5¼	Amiga-beige	beige	479,—	40/80 Tracks
Kröning Computersysteme	k.A.	nein	nein	NEC 1036	3½	k.A.	k.A.	349,—	—
Microware	Metall	ja	nein	NEC 1036	3½	Amiga-beige	beige	399,—	—
Padercomp	k.A.	nein	nein	NEC 1036	3½	k.A.	beige	398,—	—
ProMicro GmbH	Plastik	ja	nein	NEC 1035	3½	grau	schwarz	448,—	—
Soyka Datentechnik	k.A.	nein	ja	Teac	5¼	k.A.	k.A.	448,—	40/80 Tracks
	Plastik	nein	ja	NEC 1036	2 x 3½	Amiga-beige	beige	649,—	—
	Plastik	nein	ja	NEC 1036	2 x 3½	Amiga-beige	beige	599,—	Bausatz
	Plastik	nein	ja	NEC 1036	3½	Amiga-beige	beige	324,90	Bausatz
	k.A.	nein	ja	NEC 1036	3½	k.A.	beige	289,—	—
	Plastik	nein	ja	NEC 1036	3½	Amiga-beige	beige	369,—	—
Stalter Computerbedarf	k.A.	nein	ja	NEC 1036	3½	k.A.	k.A.	299,—	—
	k.A.	nein	ja	Teac	5¼	k.A.	k.A.	399,—	40/80 Tracks
SWS Computersysteme	Metall	nein	nein	NEC 1036	3½	Amiga-beige	k.A.	389,—	—
Tröps Computertechnik	k.A.	nein	nein	NEC 1036	3½	k.A.	k.A.	269,—	ohne Gehäuse
	k.A.	nein	nein	NEC 1035	3½	k.A	k.A.	299,—	Bausatz, Fertiggerät 369 Mark
	k.A.	nein	nein	NEC 1036	3½	k.A.	k.A.	325,—	Bausatz, Fertiggerät 398 Mark
Vesalia Versand	Metall	nein	ja	NEC 1035	3½	Amiga-beige	schwarz	299,—	—
	Metall	ja	ja	NEC 1036	3½	Amiga-beige	schwarz	349,—	—
	Metall	ja	ja	NEC 1036	2 x 3½	Amiga-beige	beige	648,—	—
	Metall	ja	ja	Teac	5¼	Amiga-beige	beige	449,—	40/80 Tracks
	Metall	ja	ja	Chinon	3½	Amiga-beige	beige	329,—	—

Tabelle. Übersicht über Zusatzlaufwerke für alle Amiga-Modelle

AB-Computersysteme, Wildenburgstr. 21, 5000 Köln 41, Tel. 0221/4301442
CompWare, Schlägel & Eisenstr. 9, 4352 Herten-Langenbochum, Tel. 02366/55891
ASH Computerware, Von-Brandl-Str. 15, 8229 Laufen/Salzach, Tel. 08682/1327
B. Soltau Electronics, Esplanade 39, 2000 Hamburg 36, Tel. 040/340445 — 340394
C.A.S., Sprendlinger Landstr. 71, 6050 Offenbach, Tel. 069/842013
CompTec, Obermörmtererstr. 1, 4192 Kalkar 4, Tel. 02824/3867
Robert Daam Computer, Postfach 702, 8950 Kaufbeuren 2, Tel. 08341/81489
Datentechnik Bittendorf, Postfach 100248, 6360 Friedberg, Tel. 06031/61950
Dolphin Software, Egenolfstr. 19, 6000 Frankfurt 1, Tel. 069/446573
Flesch & Hörnemann, Schlägel & Eisenstr. 46, 4352 Herten, Tel. 02366/55176
Futurevision, Fr.-Veith-Str. 21, 6128 Höchst, Tel. 06163/1278
Interplan, Nymphenburger Str. 134, 8000 München 19, Tel. 089/1234066
Kupke Computertechnik, Apelank 28, 4600 Dortmund 16, Tel. 0231/852605
Kröning Computersysteme, Deichsberg 2, 4790 Paderborn, Tel. 05254/69369
Microware, Salmdorf 2, 8013 Haar, Tel. 089/4391096
Padercomp, Erzbergerstr. 27, 4790 Paderborn, Tel. 05251/36396
ProMicro GmbH, Teglerstr. 6, 1000 Berlin 65, Tel. 030/4653088
Soyka Datentechnik, Hattingerstr. 685, 4630 Bochum, Tel. 0234/411913
Stalter Computerbedarf, Kurt-Schumacher-Str. 6, 6750 Kaiserslautern, Tel. 0631/24285
SWS Computersysteme, Am Rathaus 8, 8395 Hauzenberg b. Passau, Tel. 08586/2174
Tröps Computertechnik, Pingsdorferstr. 141, 5040 Brühl, Tel. 02232/47105
Vesalia Versand, Marienweg 40, 4230 Wesel, Tel. 0281/65466 — 62205

Genug Platz für alle

Für viele Anwendungen sind 512 KByte Hauptspeicher zu wenig. Eine Aufrüstung auf mehr RAM tut not. Die folgende Vorstellung hilft Ihnen, sich für die richtige Speichererweiterung zu entscheiden.

Ein Multitasking-Computer wie der Amiga erlaubt es dem Anwender, mehrere Programme parallel im Speicher zu halten und jederzeit damit zu arbeiten. Doch mit 512 KByte ist oft nicht genügend Hauptspeicher vorhanden, um diese nützliche Einrichtung auch in Anspruch zu nehmen. Gerade beim Amiga ist eine RAM-Erweiterung in vielerlei Hinsicht zu überdenken: Seien es Anwendungen wie Datenbank-Systeme oder Textverarbeitungen, Grafikprogramme oder einfach nur Reserven. 1 MByte müßten schon sein, besser wären 2 MByte oder mehr. Der folgende Artikel stellt Ihnen empfehlenswerte Speichererweiterungen vor und gibt Ihnen einen tabellarischen Überblick über das derzeitige Marktangebot.

Derzeit sind die meisten RAM-Erweiterungen noch für den Amiga 1000 konzipiert. So auch die 1 MByte-Box von C-Ltd. (Bild 1), die zum Preis von 1195 Mark von DTM und Intelligent Memory vertrieben wird. Die in einem stabilen Metallgehäuse untergebrachte Platine ist autokonfigurierend und besitzt einen durchgeschleiften Bus, so daß mehrere Geräte aneinandergesteckt werden können. Die Box ist problemlos am A1000 zu betreiben und nimmt nicht viel Platz in Anspruch. Zwar liegt der Preis für 1 MByte etwas hoch, doch kann die solide Verarbeitung und der tadellose Kundenservice dies wieder wettmachen.

Ebenfalls für den Amiga 1000 gedacht ist die Comspec-Box (Bild 2), die von DTM, Amigaland und Intelligent Memory angeboten wird. Das sehr solide verarbeitete Metallgehäuse enthält eine Platine, die dem Computer ohne Probleme zu 2 MByte mehr RAM verhilft. Sie ist autokonfigurierend und auch der Bus ist erfreulicherweise durchgeschleift. Es können also noch andere Geräte mit der Box am Expansion-Port betrieben werden. Der Preis, der zwischen 1198 und 1498 Mark liegt, ist in Relation zu dem, was man beim Kauf erhält, als fair zu bezeichnen.

Sollten 2 MByte nicht genug sein, hilft das RAM-Board von CompuStore zu höheren Dimensionen. Das 1848 Mark teuere, autokonfigurierende Grundgerät (Bild 3), das über einen durchgeschleiften Bus verfügt, vergrößert den Hauptspeicher um 2 MByte. Als Zusatz gibt es 650 Mark teure Steckkarten, die den Speicher um jeweils 2 MByte bis zu maximal 8 MByte aufrüsten. Das abschaltbare RAM-Board ist problemlos zu betreiben und belastet durch ein eingebautes Netzteil nicht unnötig den Computer. Das große Gerät ist etwa so lang und breit wie der A1000, so daß es auf den Computer gestellt werden kann.

Etwas Spezielles stellt die 512-KByte-Erweiterung von Tröps dar (Bild 4). Sie kann wahlweise mit RAMs oder EPROMs bestückt werden. Die Karte wird seitlich an den Expansion-Port des A1000 angesteckt und muß mit dem CLI-Befehl ADDMEM zugewiesen werden. Sie besitzt einen durchgeschleiften Bus, aber kein Gehäuse. Die Karte ist wohl eher etwas für Bastelfreaks, da sie sowohl als Bausatz wie auch als unbestücktes Fertiggerät angeboten wird.

Für den Proficomputer Amiga 2000 ist leider derzeit nicht viel zu haben. Die 2-MByte-Karte von Commodore (Bild 5), die für etwa 950 Mark im Fachhandel angeboten wird, ist derzeit das einzige, was in einen der freien Amiga-Slots zur

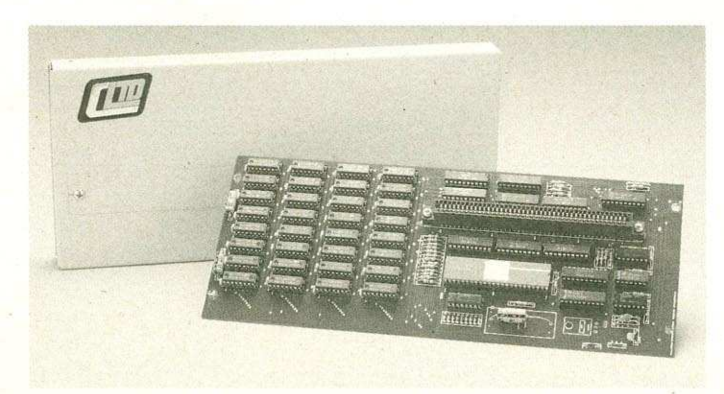


Bild 1. Sorgt für 1 MByte mehr Speicher: die C-Ltd.-Erweiterung für den Amiga 1000 von Intelligent Memory

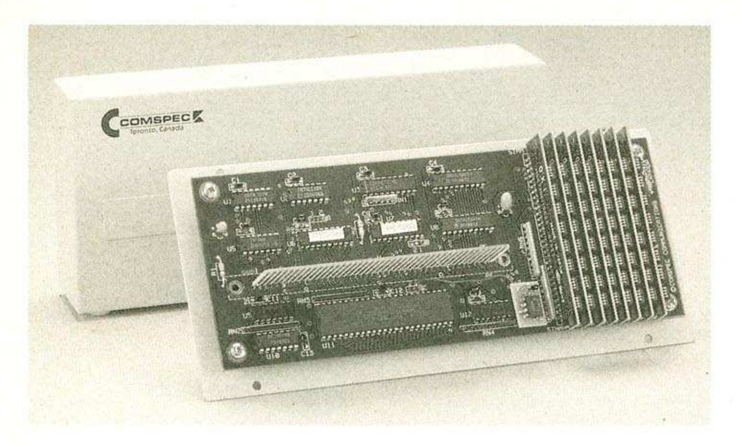


Bild 2. Comspec von DTM: einfach anstecken und der Amiga 1000 verfügt über 2 MByte mehr Speicher

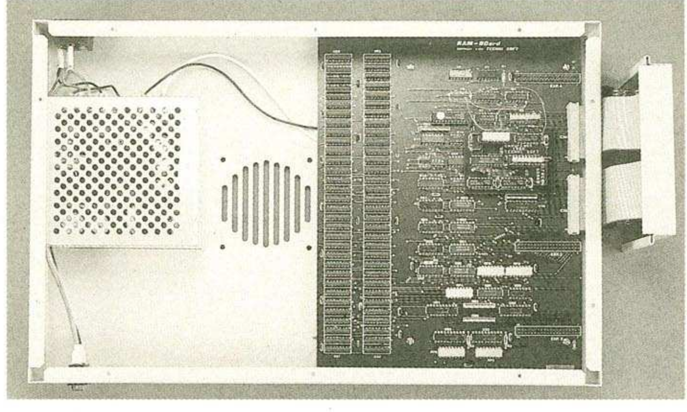


Bild 3. Durch zusätzliche Karten erweiterbar bis auf 8 MByte: das voluminöse RAM-Board von CompuStore

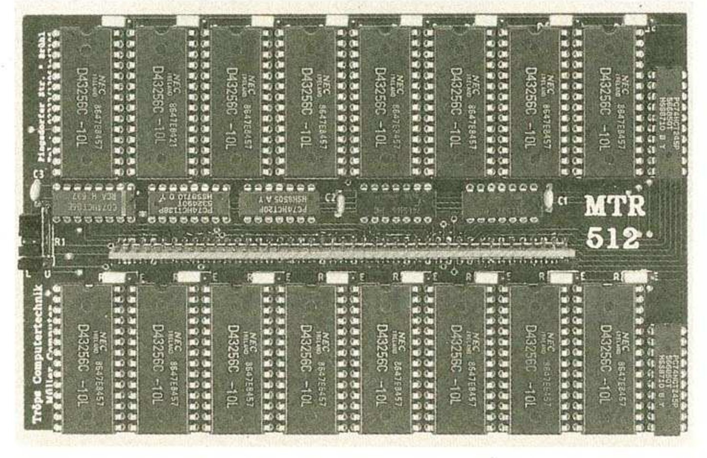


Bild 4. Die 512-KByte-Karte von Tröps kann wahlweise sowohl mit RAMs als auch mit EPROMs bestückt werden

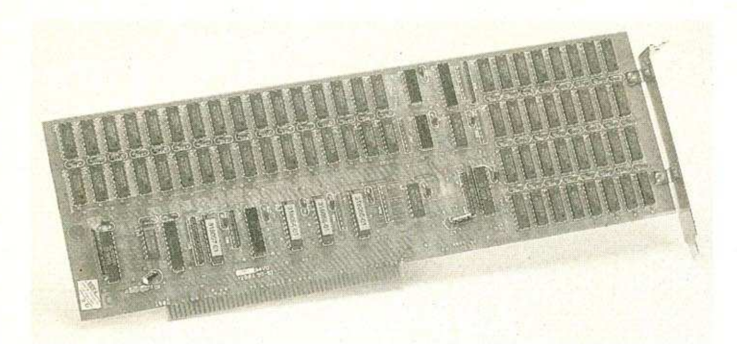


Bild 5. Wertet den Amiga 2000 um 2 MByte auf: die RAM-Karte von Commodore zum internen Einbau

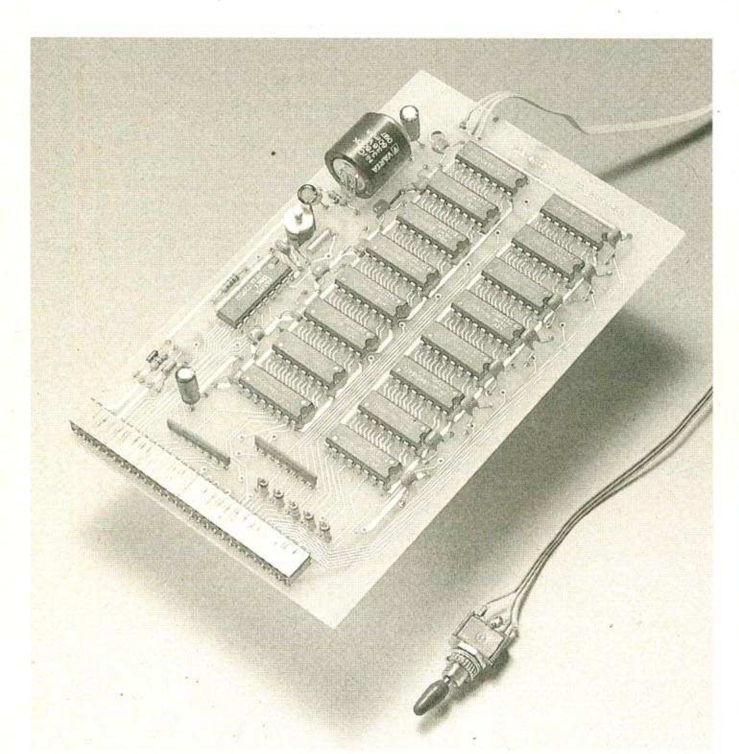


Bild 6. Bitter nötige Erweiterung für den A500: abschaltbare 512 KByte von Bittendorf mit eingebauter Echtzeituhr

Speichervergrößerung eingesetzt werden kann. Die Karte ist selbstverständlich autokonfigurierend und wird problemlos vom A2000 akzeptiert. Es können auch mehrere dieser Erweiterungen in die Slots gesteckt werden, so daß sich der Computer auf über 8 MByte Hauptspeicher erweitern läßt.

Mehr Speicher bitter nötig hat auch der Amiga 500. Eine Möglichkeit, ihm mehr RAM zukommen zu lassen, ist der Einbau einer kleinen Karte, die über eine batteriegepufferte Echtzeituhr verfügt und 512 KByte zusätzliches RAM bietet. Die kleinen Karten werden an der Unterseite des A500 eingesteckt, wo sie unter der Abdeckklappe verschwinden. Vorteilhaft ist es, wenn sich die Karten abschalten lassen, zwingend erforderlich ist es aber nicht. Die Bilder 6 und 7 zeigen Ihnen verschiedene A500-Erweiterungen, wie sie zu einem Preis von etwa 250 Mark angeboten werden.

Aber beim A500 geht auch mehr: C.A.S. verkauft seit kurzer Zeit eine speziell für diesen Computer angefertigte, autokonfigurierende Erweiterung (Bild 8). Mit 2 MByte ist sie zu einem guten Preis von 895 Mark erhältlich. Die Box, die direkt an den A500 angesteckt werden kann, besteht aus solidem Blech und ist abschaltbar. Sie ist eine empfehlenswerte Alternative für unterdimensionierte Amiga 500-Modelle.

Wer die Erweiterungen lieber dezent im Computer »verstecken« möchte, kann dies mit Karten tun, die in den Computer eingebaut werden. Typisches Beispiel dafür ist die 1-MByte-Erweiterung von Futurevision für den Amiga 1000, die mit einer mitgelieferten, angepaßten Kickstart-Version 749 Mark kostet (Bild 9). Wenn andere (leider derzeit sehr teurere) RAM-Bausteine eingesetzt werden, läßt sich der Speicher sogar um 4 MByte aufstocken. Allerdings muß für diese Art von Speichererweiterungen der Amiga aufgeschraubt werden, was Garantieverlust bedeutet. Die Erweiterung ist aber relativ unkompliziert einzubauen, da sie zwischen dem 68000-Prozessor und dessen Sockel eingesteckt wird.

Eine andere Karte, die einen Standard-A1000 auf insgesamt 1 MByte aufrüstet, ist die etwas ungewöhnliche Erweiterung von C.A.S. (Bild 10). Sie wird in die Frontaussparung eingesteckt. Allerdings müssen einige Lötarbeiten in Inneren des Amiga vogenommen werden, womit diese Karte sicher nichts für »Lötkolbengegner« ist. C.A.S. bietet aber einen Einbauservice an. Die Speichererweiterung ist nicht autokonfigurierend, eine mitgelieferte Kickstart-Diskette installiert den zusätzlichen Speicher aber ordnungsgemäß.

Welche Wahl?

Egal, für welche RAM-Erweiterung Sie sich entscheiden — eine sinnvole Ergänzung zu Ihrem Computersystem ist es allemal. Einzig bei der Kapazität der Geräte und der Art des Einbaus sollte überlegt werden, für welches Angebot die Entscheidung getroffen wird.

In der abschließenden Tabelle sind zum besseren Überblick alle Speichererweiterungen aufgeführt, zu denen uns Informationen vorlagen. Es ist beschrieben, ob der Einbau intern oder extern erfolgt, wel-

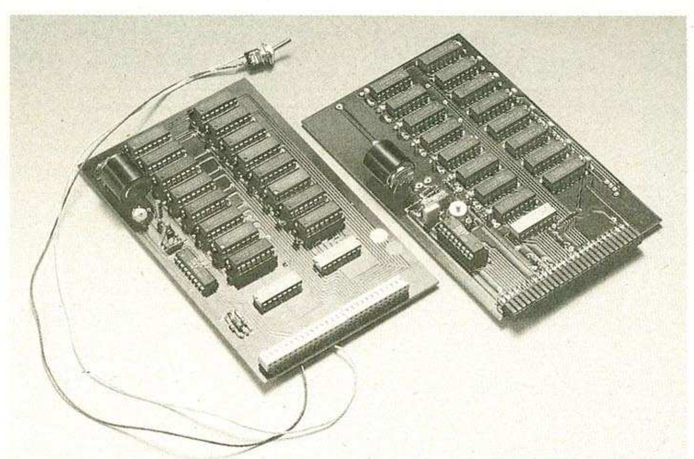


Bild 7. Ebenfalls 512 KByte für den A500 von Vesalia Versand (links) und C.A.S. (rechts) zum internen Einbau

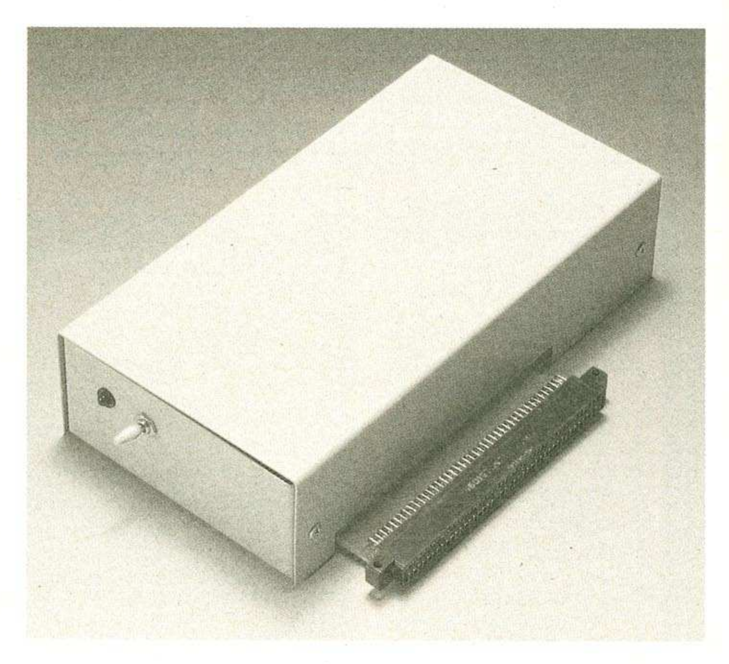


Bild 8. Derzeit die einzige 2-MByte-Speichererweiterung für den Amiga 500 zum externen Anbau von C.A.S.

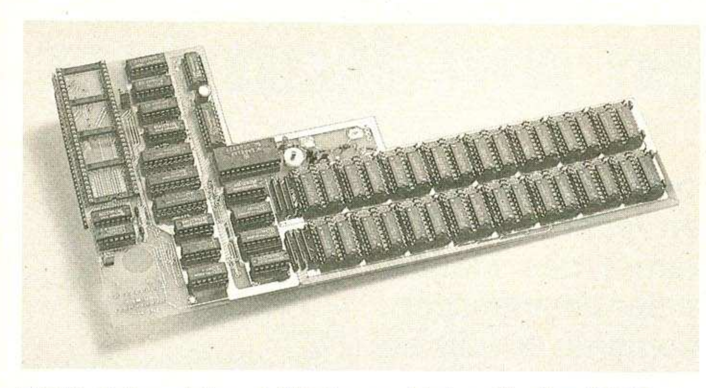


Bild 9. Futurevision: 1 MByte zum Einbau für den A1000

cher Computertyp Voraussetzung ist und mit welchen Kapazitäten die Erweiterungen erhältlich sind. Sollten Besonderheiten dabeisein, sind diese in der Spalte »Bemerkungen« eingetragen. Die Adressen der Anbieter schließen die Übersicht über die sicher sinnvollen Ergänzungen Ihres Arbeitsgeräts ab. (dm)

Alcomp, Lessingstr. 46, 5012 Bedburg, Tel. 02272/1580
Alphatron, Luitpoldstr. 22, 8520 Erlangen, Tel. 09131/25018
C.A.S., Sprendlinger Landstr. 71, 6050 Offenbach, Tel. 069/842013
CompTec, Obermörmtererstr. 1, 4192 Kalkar 4, Tel. 02824/3867
CompuStore, Fritz-Reuter-Str. 6, 6000 Frankfurt/Main 1, Tel. 069/567399
Datentechnik Bittendorf, Postfach 100248, 6360 Friedberg, Tel. 06031/61950
Dolphin Software, Egenolfstr. 19, 6000 Frankfurt 1, Tel. 069/446573
DTM, Poststr. 25, 6200 Wiesbaden, Tel. 06121/560084
Flesch & Hörnemann, Schlägel & Eisenstr. 46, 4352 Herten, Tel. 02366/55176
Futurevision, Fr.-Veith-Str. 21, 6128 Höchst, Tel. 06163/1278
Intelligent Memory, Basaltstr. 58, 6000 Frankfurt 90, Tel. 069/707 1102
Kupke Computertechnik, Apelank 28, 4600 Dortmund 16, Tel. 0231/852605
M. Kröning Computersysteme, Deichsberg 2, 4790 Paderborn, Tel. 05254/69369
Merkens EDV, Fuchstanzstr. 6a, 6231 Schwalbach, Tel. 06196/3026
Computersysteme M. Lamm, Schönbornring 14, 6078 Neu-Isenburg 2, Tel. 06102/52535
Softwareland, Franklinstr. 27, CH-8050 Zürich, Tel. 0041/1-3115959
Soyka Datentechnik, Hattingerstr. 685, 4630 Bochum 5, Tel. 0234/411913
Tröps Computertechnik, Pingsdorferstr. 141, 5040 Brühl, Tel. 02232/47105
Vesalia Versand, Marienweg 40, 4230 Wesel, Tel. 0281/65466 — 62205

Die Übersicht erhebt keinen Anspruch auf Vollständigkeit. Mögliche Preisänderungen nach Redaktionsschluß konnten nicht berücksichtigt werden.

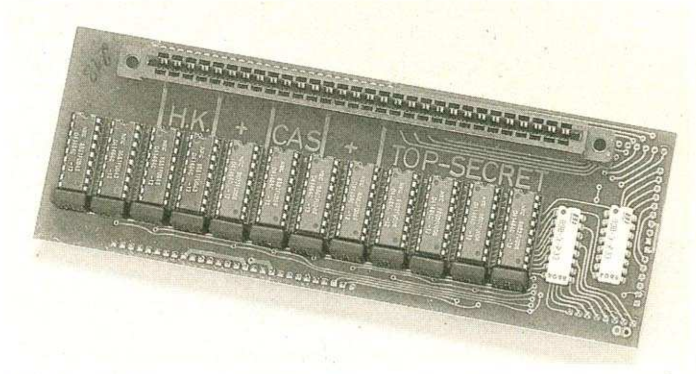


Bild 10. C.A.S. wertet den A1000 um 768 KByte auf

Anbieter/Gerät	Einbau	geeignet für	max. Kapazität	bestückt mit	mit Gehäuse	autokonfigurierend	Bus durchgeführt	Preis in Mark	Bemerkung
Alcomp/A500	Intern	A500	512 KByte	512 KByte	nein	ja	—	189,—	ohne Uhr
Alphatronic/Dram-Ex	Intern	A1000	4 MByte	1 MByte	nein	nein	—	998,—	mit Echtzeituhr 1173 Mark
C.A.S./A500-512 KB	Intern	A500	512 KByte	512 KByte	nein	ja	—	239,—	mit Echtzeituhr
/A500-512 KB	Intern	A1000	768 KByte	768 KByte	Frontabdeckung	modifizierter Kickstart	—	378,—	—
/A500-2 MB	Extern	A500	2 MByte	2 MByte	ja	ja	nein	895,—	abschaltbar
/A1000-2 MB	Extern	A1000	2 MByte	2 MByte	ja	ja	ja	895,—	abschaltbar
CompTec/									
A500-512 KB	Intern	A500	512 KByte	512 KByte	nein	ja	—	199,—	mit Echtzeituhr 249 Mark
A1000-2 MB	Intern	A1000	4 MByte	2 MByte	nein	modifizierter Kickstart	—	888,—	mit Echtzeituhr
A1000-2 MB ext.	Extern	A1000	2 MByte	2 MByte	ja	ja	ja	1095,—	abschaltbar
A2000-RAM	Intern	A2000	4 MByte	2 MByte	nein	ja	—	956,—	—
CompuStore/RAM-BOard	Extern	A1000	8 MByte	2 MByte	ja	ja	ja	1848,—	je 2 MB zusätzlich 650 Mark
Bittendorf/									
A500-512 KB	Intern	A500	512 KByte	512 KByte	nein	ja	—	222,—	abschaltbar, mit Echtzeituhr
RAM-Chips	Intern	A2000	512 KByte	512 KByte	—	—	—	199,—	Chipsatz für 1-MB-Karte
A2000-2 MB	Intern	A2000	2 MByte	2 MByte	nein	ja	—	949,—	Commodore-Erweiterung
Dolphin/									
A1000-256 K	Intern	A1000	256 KByte	256 KByte	nein	ja	—	129,—	Commodore-Erweiterung
512 MB	Extern	A1000	2 MByte	1 MByte	ja	ja	ja	895,—	2 MB = 1295 Mark
DTM/Comspec	Extern	A1000	2 MByte	2 MByte	ja	ja	ja	1498,—	—
Futurevision/A1000 int.	Intern	A1000	4 MByte	1 MByte	nein	modifizierter Kickstart	—	749,—	2 MB = 1248 Mark, 4 MB = 2048 Mark
A1000 ext.	Extern	A1000	2 MByte	2 MByte	ja	ja	ja	998,—	abschaltbar
A500-512 KB	Intern	A500	512 KByte	512 KByte	nein	ja	—	258,—	mit Echtzeituhr
Flesch & Hörnemann/A500	Intern	A500	1,5 MByte	1,5 MByte	nein	ja	—	998,—	—
Intelligent Memory/C-Ltd.	Extern	A1000	2 MByte	1 MByte	ja	ja	ja	1195,—	—
Kupke/Golem	Extern	A1000	2 MByte	2 MByte	ja	ja	ja	998,—	abschaltbar
Kröning/A1000-RAM	Intern	A1000	4 MByte	1 MByte	nein	nein	—	749,—	2 MB = 1249 Mark, 4 MB = 2049 Mark
Merkens/2 MB	Extern	A1000	2 MByte	2 MByte	ja	ja	k.A.	1138,—	—
2 MB-A2000	Intern	A2000	4 MByte	2 MByte	nein	ja	—	948,—	—
M. Lamm/MegaMiga	Intern	A1000	1 MByte	1 MByte	nein	modifizierter Kickstart	—	498,—	—
Softwareland/Alegra	Extern	A1000	2 MByte	512 KByte	nein	ja	—	835,—	2 MB = 1649 Mark
Insider	Intern	A1000	1 MByte	1 MByte	nein	ja	—	814,—	—
Soyka/Amiga RAM	Intern	A500	512 KByte	512 KByte	nein	ja	—	275,—	mit Uhr, abschaltbar = 289 Mark
Tröps/MTR 512	Extern	A1000	512 KByte	512 KByte	nein	nein	ja	598,—	EPROMs anstelle RAM möglich
Vesalia/A500-512 KB	Intern	A500	512 KByte	512 KByte	nein	ja	—	249,—	mit Echtzeituhr, abschaltbar = 259 Mark

DCopy – Kopieren mit Komfort

Das Programm »Diskcopy« von der Workbench ist zum Kopieren recht unkomfortabel. Bei dem Programm des Monats können Sie viele Parameter beeinflussen. Auch das mehrfache Kopieren auf bis zu drei Laufwerke beherrscht »DCopy«.

Das in der Programmiersprache C geschriebene Programm DCopy läßt beim Kopieren keine Wünsche mehr offen. Bis zu vier Laufwerke erkennt DCopy automatisch und benutzt sie auch bei Bedarf. Leicht können die gewünschten Parameter eingestellt werden. Die Eingaben erfolgen über die Tastatur, wobei die Cursor-Steuerung vorbildlich gelöst ist.

DCopy öffnet ein eigenes Fenster, wodurch es auch von der Workbench aus gestartet werden kann. Nach dem Aufruf erscheinen in dem Fenster die voreingestellten Werte (siehe Bild 1).

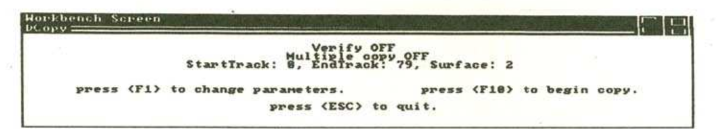


Bild 1. Das Hauptmenü von »DCopy« mit der Anzeige der aktuellen Parameter wie Verify und Multiplecopy

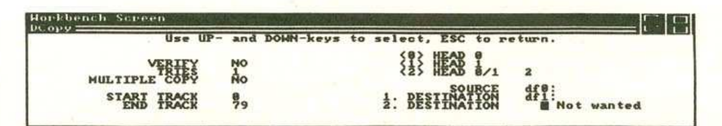


Bild 2. In diesem Untermenü von »DCopy« können Sie die Parameter für den Kopiervorgang verändern

In der ersten Zeile steht nach dem Starten »Verify OFF«. Dies bedeutet, daß nach dem Schreiben der Daten keine Überprüfung auf Richtigkeit durchgeführt wird. In diesem Modus erhöht sich natürlich die Kopiergeschwindigkeit, da die Daten nicht noch einmal gelesen werden müssen.

Darunter steht, ob das Mehrfachkopieren eingeschaltet ist (»Multiple copy OFF«). Ist dieser Modus eingeschaltet, können mehrere Kopien erstellt werden, wobei die Daten nur einmal geladen werden. Auch hierbei ist es möglich, auf mehrere Zieldisketten in einem Durchlauf zu kopieren.

Sehr interessant ist die Fähigkeit von DCopy, nicht nur die gesamte Diskette, sondern auch beliebige Teile davon zu duplizieren. »StartTrack« und »EndTrack« sind die Spuren, von beziehungsweise bis zu denen der Kopiervorgang laufen soll. Mit »Surface« gibt man an, welche Diskettenseite vervielfältigt werden soll. Auf die Bedeutung dieses Parameters gehen wir bei der Beschreibung des Untermenüs ein.

Durch Drücken von <F1> gelangen Sie in ein Untermenü (siehe Bild 2), in dem die voreingestellten Werte editierbar sind. Um zu dem Parameter zu gelangen, den Sie ändern wollen, benutzen Sie die Tasten <Cursor hoch> und <Cursor runter>.

Bei »VERIFY« werden nur die Tasten <y> und <n> akzeptiert, die Verify an- beziehungsweise ausschalten.

Mit »TRIES« wird bestimmt, wie viele Schreibversuche unternommen werden, wenn ein Schreibfehler auftritt. Diese Funktion ist somit nur dann sinnvoll, wenn »VERIFY« aktiviert ist.

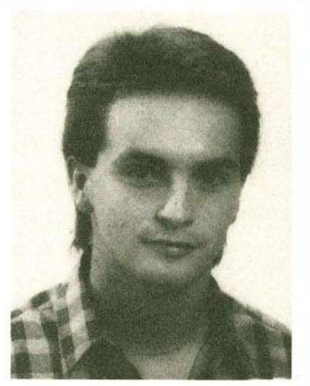
Für mehrfache Kopien bei nur einmaligem Lesen ist der Parameter »MULTIPLE COPY« zuständig. Genau wie bei VERIFY kann nur <y> oder <n> eingegeben werden. Positiv ist dieser Menüpunkt, wenn man nur über ein Laufwerk verfügt, da man zeitsparend mehrere Kopien erstellen kann.

»STARTTRACK« und »ENDTRACK« geben die Start- und Endspur für den Kopiervorgang an. Falls die Startspur größer als die Endspur ist, kann man das Menü nicht verlassen. Um nur eine Spur zu kopieren, stellt man einfach

Fortsetzung auf Seite 104

Jeden Monat 2000 DM GEWINNEN

Gerald Fries, der Autor des Programm des Monats, begann im Jahr 1984 sich mit Computern zu beschäftigen. Sein erster Computer war ein C 64, auf dem er einige mathematische Programme erstellte. Letztes Jahr erfolgte dann der Aufstieg zum Amiga, von dessen Fähigkeiten der Autor von Anfang an begeistert war. So verwaltet er zum Beispiel mit eigenen Programmen seine Musikkassetten und die Haushaltskasse. Das Programm DCopy entstand, weil dem Autor an jedem Kopierprogramm ein Detail fehlte.



Im Augenblick studiert er im neunten Semester Elektrotechnik mit dem Schwerpunkt «Rechnergestützter Schaltungsentwurf«.

Programmname:	DCopy
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40
Aufrufe:	cc dcopy+l-s n dcopy.o-lm32-cc32

```
 1 ej0 #include <devices/console.h>
 2 Qf  #include <intuition/intuition.h>
 3 CN  #include <stdio.h>
 4 h4  #include <devices/trackdisk.h>
 5 CH  #include <exec/memory.h>
 6 4r  #include <libraries/dos.h>
 7 Oy  #include <libraries/dosextens.h>
 8 Ic  #define BLOCKSIZE 5632L        /*  BLOCKSIZE*NUMSECS  (512*11
       )  */
 9 T9  #define YES 1
10 bc  #define NO 0
11 1J  #define FAILED -10             /*  operation-failed flag  */
12 Kc  #define OK 0
13 zt  struct Process *mp,*FindTask();
14 Ok  struct MsgPort *diskport[4];
15 xM  struct IOExtTD *diskreq[4];
16 Fg  BYTE *erase,*puffmerk,*fastdiskpuffer[160],*diskpuffer[80],*
       AllocMem();
```

Listing. »DCopy«, ein komfortables Kopierprogramm für den Amiga. Bitte mit dem Checksummer (siehe S. 76) eingeben.

```
17 iP  struct IntuitionBase *IntuitionBase,*OpenLibrary();
18 zl  extern struct MsgPort *CreatePort();
19 Sa  extern struct IORequest *CreateExtIO();
20 UT  struct NewWindow nw = {
21 ft8        0, 10,                   /* starting position (left,top
              ) */
22 Iq         640,100,                 /* width, height */
23 lg         -1,-1,                   /* detailpen, blockpen */
24 cE         0,                       /* flags for idcmp */
25 c4         WINDOWDEPTH |
26 Eq         WINDOWDRAG |
27 Ot         NOCAREREFRESH |
28 GF         ACTIVATE,                /* window gadget flags */
29 N4         OL,                      /* pointer to 1st user gadget
              */
30 BH         NULL,                    /* pointer to user check */
31 Gn         (UBYTE *)"DCopy",        /* title */
32 Xo         NULL,                    /* pointer to window screen */
33 qd         NULL,                    /* pointer to super bitmap */
34 GE         0,0,                     /* min width, height */
35 XX         0,0,                     /* max width, height */
36 ix         WBENCHSCREEN};
37 pv0 struct Window *w,*OpenWindow();
38 1a  struct IOStdReq *WMsg;                    /* I/O request block
    pointer */
39 bB  struct MsgPort *consoleWritePort;       /* a port at which to
     receive */
40 JD  struct IOStdReq *RMsg;                    /* I/O request block
    pointer */
41 wX  struct MsgPort *consoleReadPort;        /* a port at which to
     receive */
42 7m  extern struct IOStdReq *CreateStdIO();
43 Yr  char readstring;   /* provides a buffer even though using on
    ly one char */
44 Qq  char string[50];   /* provides a buffer for writing variable
    s */
45 QA  int error = 0;
46 U5  int avail = 0;
47 2p  int fastavail = 0;
48 Vo  int anzdisk = 0;
49 jD  /*  verify,maxtry,multiple,starttrack,endtrack,mode,source,3
    x destination  */
50 S1  int flag[10] = {'n',1,'n',0,79,2,0,-1,'n','n'};
51 VD  int t[4] = {-1,-1,-1,-1};            /*  remember OpenDevice f
    ailures  */
52 Xt  main()
53 nG  {
54 Sw3    int i,success,fail = 0;
55 Qd     long dos_packet();
56 Yk     if((IntuitionBase=OpenLibrary("intuition.library",0))==NU
       LL)
57 rK6       exit(99L);
58 UV3    consoleWritePort = CreatePort("my.con.write",0L);
59 XE     if(consoleWritePort == 0)
60 uN6       exit(99L);
61 vJ3    WMsg =  CreateStdIO(consoleWritePort);
62 aH     if(consoleWritePort == 0)
63 xQ6       exit(99L);
64 qy3    consoleReadPort = CreatePort("my.con.read",0L);
65 VU     if(consoleReadPort == 0)
66 OT6       exit(99L);
67 KK3    RMsg =  CreateStdIO(consoleReadPort);
68 MB     if(RMsg == 0)
69 3W6       exit(99L);
70 Rs3    w = OpenWindow(&nw);   /* create a window */
71 jO     if(w == NULL)
72 6Z6       exit(99L);
73 203    error = OpenConsole(WMsg,RMsg,w);
74 VU     if(error != 0)
75 9o6       exit(99L);
76 TP        /* attach a console to this window, initialize
77 vG7         * for both write and read */
78 iD3    QueueRead(RMsg,&readstring); /* tell console where to
79 9fh                                 * put a character that
80 Zq                                  * it wants to give me
81 ru                                  * queue up first read
82 Nm                                  */
83 EY3    CursorOff();                 /* cursor will not be
        displayed  */
84 3pJ                /*  How many drives are connected ?  */
85 nL3    for (anzdisk=0;anzdisk<4;anzdisk++) {
86 rr6       if ((diskport[anzdisk] = CreatePort(0L,0L)) == 0) {
87 UY9          puts("\nCan't create port.\n");
88 Zf           exit(0L);
89 Rw6       }
90 km        diskreq[anzdisk] = (struct IOExtTD *)
91 CwE               CreateExtIO(diskport[anzdisk],(long)sizeof(str
                  uct IOExtTD));
92 2x6       if (OpenDevice(TD_NAME,(long)anzdisk,diskreq[anzdisk],
```

```
93 tr9           DeleteExtIO(diskreq[anzdisk],(long)sizeof(struct IO
              ExtTD));
94 CI            DeletePort(diskport[anzdisk]);
95 EC            t[fail++]=anzdisk+'0';
96 Y36        }
97 Ka         else {
98 UK9           if (flag[7]==-1 && anzdisk!=0)
99 FUC              flag[7]=anzdisk;
100 D4K                  /* do not verify disk-changes */
101 CB9          mp=FindTask("File System");
102 m0           dos_packet(&mp->pr_MsgPort,ACTION_INHIBIT,TRUE,0L,0
             L,0L,0L,0L,0L);
103 fA6       }
104 gB3    }
105 nL     anzdisk-=fail;              /*  amount of connected drives
         */
106 qP     if (anzdisk==1)             /*  only one drive connected
       */
107 1q6       flag[7]=0;
108 6w3    do {
109 ra6       error = 0;
110 q7        if (flag[0]=='y') {
111 Qh9          if (flag[1]==1) {
112 HOC             locate (2,30);
113 DJ              sprintf(string,"Verify ON with %d try",flag[1]);
114 2y              ConPutStr(WMsg,string);
115 rM9          }
116 dt           else {
117 1OC             locate (2,29);
118 3I              sprintf(string,"Verify ON with %d tries",flag[1]
                );
119 73              ConPutStr(WMsg,string);
120 wR9          }
121 xS6       }
122 jz        else {
123 tu9          locate(2,35);ConPutStr(WMsg,"Verify OFF");
124 OV6       }
125 NP        locate(3,32);ConPutStr(WMsg,"Multiple copy ");
126 xd        if (flag[2]=='y')
127 PR9          ConPutStr(WMsg,"ON ");
128 tg6       else
129 zV9          ConPutStr(WMsg,"OFF");
130 N16       locate(4,20);
131 gy        sprintf(string,"StartTrack: %d, EndTrack: %d, Surface:
          %d",
132 sYM                      flag[3],flag[4],flag[5]);
133 LH6       ConPutStr(WMsg,string);
134 hf        locate (7,7);ConPutStr(WMsg,"press <F1> to change
          parameters.");
135 qs        locate (7,48);ConPutStr(WMsg,"press <F10> to begin cop
          y.");
136 jf        locate (9,30);ConPutStr(WMsg,"press <ESC> to quit.");
137 Vg        i=Taste();
138 8C        if (i==155)
139 VL9          i=ConGetChar(consoleReadPort,RMsg,&readstring);
140 Yo6       if (i==48) {
141 dZ9          ConGetChar(consoleReadPort,RMsg,&readstring);
142 qq           success=Menu();
143 Jo6       }
144 QT        else if (i==57) {
145 Kf9          locate(1,2);
146 kr           ConPutStr(WMsg,"
                        "
147 uNO                     "                          ");
148 MF9          locate (7,7);ConPutStr(WMsg,"
                           ");
149 6q           locate(7,48);ConPutStr(WMsg,"
                    ");
150 Rz           locate(9,30);ConPutStr(WMsg,"
                 ");
151 nJ           ConGetChar(consoleReadPort,RMsg,&readstring);
152 Qi           success=Copy();
153 Ty6       }
154 zW        else if (i==27)
155 FB9          End();
156 qk6       CursorOff();
157 BF        ConPutStr(WMsg,"\f");
158 6f        if (success==FAILED) {
159 oU9          locate(1,25);
160 ox           ConPutStr(WMsg,"\x9b0;33;40mATTENTION:     Copy fai
             led !!!\x9bm");
161 b66       }
162 ZJ3    } while (1<2);   /*  for ever !!  */
163 d80 }
164 OU  Menu()
165 b4  {
166 pJ3    int zeile = 8;
167 9h     int i,k,l,spalte,j;
168 zc     int success = OK;
169 P3     ConPutStr(WMsg,"\f");CursorOn();
```

```
170 Oc      locate(1,17);ConPutStr(WMsg,"Use UP- and DOWN-keys to sel
            ect, "
171 Y5V                              "ESC to return.");
172 rz3     locate (4,15);ConPutStr(WMsg,"VERIFY");locate(4,25);
173 WK      if (flag[0]=='y')
174 B36        ConPutStr(WMsg,"YES");
175 eR3     else
176 DF6        ConPutStr(WMsg,"NO ");
177 fT3     locate (5,16);ConPutStr(WMsg,"TRIES");
178 JK      locate(5,25);sprintf(string,"%d",flag[1]);ConPutStr(WMsg,
            string);
179 zB      locate (6,8);ConPutStr(WMsg,"MULTIPLE COPY"),locate(6,25)
            ;
180 pV      if (flag[2]=='y')
181 IA6        ConPutStr(WMsg,"YES");
182 1Y3     else
183 KM6        ConPutStr(WMsg,"NO ");
184 1X3     locate(8,10);ConPutStr(WMsg,"START TRACK");
185 ms      locate(8,25);sprintf(string,"%d",flag[3]);ConPutStr(WMsg,
            string);
186 Nr      locate(9,12);ConPutStr(WMsg,"END TRACK");
187 x5      locate(9,25);sprintf(string,"%d",flag[4]);ConPutStr(WMsg,
            string);
188 VX      locate(3,45);ConPutStr(WMsg,"<0> HEAD 0");
189 hm      locate(4,45);ConPutStr(WMsg,"<1> HEAD 1");
190 nM      locate(5,45);ConPutStr(WMsg,"<2> HEAD 0/1");
191 4o      locate(5,60);sprintf(string,"%d",flag[5]);ConPutStr(WMsg,
            string);
192 JB      locate(7,51);ConPutStr(WMsg,"SOURCE");
193 PX      locate(7,60);sprintf(string,"df%d:",flag[6]);ConPutStr(WM
            sg,string);
194 Ec      k=1;
195 VL      do {
196 Sg6        locate(zeile,43);sprintf(string,"%d. DESTINATION",k);
197 NJ         ConPutStr(WMsg,string);
198 ze         if (k>1 && (flag[6+k]=='n'||flag[6+k]==62)) {
199 iS9           locate(zeile,60);
200 ND            ConPutStr(WMsg,"    Not wanted      ");
201 Fk6        }
202 1H         else {
203 mW9           locate(zeile,60);
204 Us            sprintf(string,"df%d:",flag[k+6]);ConPutStr(WMsg,st
              ring);
205 Jo6        }
206 i1         zeile++;
207 M83     } while (++k<anzdisk);
208 DF      /* Editor */
209 Lg      i=0;
210 6S      for (;;) {
211 vz0 start:
212 Ha6        switch (i) {
213 Rh9           case 0:
214 qHC              zeile=4;spalte=25;break;
215 Vm9           case 1:
216 avC              zeile=5;break;
217 Zr9           case 2:
218 f1C              zeile=6;break;
219 dw9           case 3:
220 nBC              zeile=8;break;
221 h19           case 4:
222 DjC              zeile=9;spalte=25;break;
223 169           case 5:
224 wNC              zeile=5;spalte=60;break;
225 pB9           case 6:
226 AfC              zeile=7;spalte=62;break;
227 tG9           case 7:
228 F1C              zeile=8;spalte=62;break;
229 xL9           case 8:
230 KrC              zeile=9;spalte=62;break;
231 1Q9           case 9:
232 9SC              zeile=10;spalte=62;break;
233 1G6        }
234 aR         locate (zeile,spalte);
235 5w         j=ConGetChar(consoleReadPort,RMsg,&readstring);
236 mn         if (j!=155) {
237 U39           if (j==66 || j==13)                    /* CRSR down,
               <CR> */
238 caC              i++;
239 WS9           else if (j==65)                        /* CRSR up *
              /
240 suC              i--;
241 1k9           else if (j==27) {                      /* ESC */
242 biC              if (flag[3] > flag[4]) {            /* starttrack
                  > endtrack? */
243 9YF                 i=4;
244 OW                  goto start;
245 xSC              }
246 CI               else if (flag[6]==flag[7]) {        /* source=de
                 stination */
247 H4F                 for (j=2;j<4;j++)                /* unselect ot
                    her drives */
248 wRI                    flag[6+j]='n';
249 MvF                 return(OK);
250 2XC              }
251 cb               else if (anzdisk>2) {      /* more than two driv
                 es connected */
252 KvF                 for (j=6 ; j<6+anzdisk ; j++)
253 ucI                    for (l=1 ; l<4 ; l++)
254 x9L                       if (flag[j]!='n' && flag[j]!=62 && l+j<
                          10)
255 se0                          if (flag[j]==flag[l+j]) {
256 iQR                             i=l+j;
257 bj                              goto start;
258 AfO                          }
259 BgC              }
260 X6               return(OK);
261 Di9           }
262 zF            else {
263 nLC              if (i==1) {
264 bnF                 if (j>'0' && j<='9') {
265 FWI                    flag[i]=j-'0';              /* integer
                       */
266 kS                     sprintf(string,"%d",flag[i]);
267 VR                     ConPutStr(WMsg,string);
268 KpF                 }
269 Ax                  else
270 MOI                    i--;
271 NsC              }
272 1Q               else if (i==5) {
273 d4F                 if (j>='0' && j<='2') {
274 OfI                    flag[i]=j-'0';              /* integer
                       */
275 tb                     sprintf(string,"%d",flag[i]);
276 ea                     ConPutStr(WMsg,string);
277 TyF                 }
278 J6                  else
279 VXI                    i--;
280 W1C              }
281 ia               else if (i==6 || i==7) {        /* 1st & 2nd driv
                 e */
282 JSF                 if(j>='0'&&j<'4'&&j!=t[0]&&j!=t[1]&&j!=t[2]&&
                    j!=t[3]) {
283 XoI                    flag[i]=j-'0';              /* integer
                       */
284 2k                     sprintf(string,"%d",flag[i]);
285 nj                     ConPutStr(WMsg,string);
286 c7F                 }
287 32                  else                           /* wrong in
                    put */
288 egI                    i--;
289 fAC              }
290 OS               else if (i>7) {                 /* 3rd & 4th d
                 rive */
291 sQF                 if((j>='0'&&j<'4'&&j!=t[0]&&j!=t[1]&&j!=t[2]&
                    &j!=t[3])
292 zsJ                   || j=='n') {    /* j != unconnected drives
                       */
293 KyI                    flag[i]=j-'0';
294 c8                     if(flag[i]==62) {
295 GOL                       locate(zeile,60);
296 vl                        ConPutStr(WMsg,"    Not wanted      ");
297 nII                    }
298 Zp                     else {
299 K4L                       locate(zeile,60);
300 B4                        sprintf(string,"df%d:    ( or 'n')",fla
                          g[i]);
301 3z                        ConPutStr(WMsg,string);
302 sNI                    }
303 tOF                 }
304 JY                  else i--;
305 vQC              }
306 Ni               else if (i==3 || i==4) {          /* start- end
                 track */
307 kIF                 if (j>='0' && j<='9') {
308 koI                    flag[i]=10*flag[i]+j-'0';
309 Fa                     if (flag[i]>NUMCYLS-1)
310 bFL                       flag[i]=j-'0';
311 c5I                    sprintf(string,"%d  ",flag[i]);
312 EA                     ConPutStr(WMsg,string);
313 3YF                 }
314 46                  i--;
315 5aC              }
316 r7               else {
317 bRF                 if (j=='y')
318 VNI                    ConPutStr(WMsg,"YES");
319 ylF                 else
320 XZI                    ConPutStr(WMsg,"NO ");
321 MHF                 flag[i]=j;                     /* character
                     */
```

```
322 ChC            }
323 zx             i++;
324 EJ9          }
325 Fk6        }
326 K5         if (i>k+5)                          /*  bottom of
           choicetable  */
327 Fa9           i=0;
328 mk6        if (i<0)                            /*  top of cho
           icetable  */
329 ho9           i=k+5;
330 Kp3     }
331 LqO }
332 RT  int anzdest[4];
333 jv  Copy()
334 Kn  {
335 kV3    SHORT cylinder;
336 23     int firsttime;
337 vk     int i,j,k,l,verify,drive,label8,label9;
338 MR     int anz[4];
339 vl     int head;
340 az     int changeflag = NO;
341 67     long warte;
342 2P     char meldung[17];
343 rl     CursorOff();
344 td     if (!avail) {
345 3G6       for (avail=0 ; avail<80 ; avail++) {
346 k69          diskpuffer[avail]=AllocMem(BLOCKSIZE,MEMF_CHIP);
347 yn           if (diskpuffer[avail]==0)
348 XgC             break;
349 d86       }
350 Bn        for (fastavail=0 ; fastavail<170-avail ; fastavail++)
          {
351 fS9          fastdiskpuffer[fastavail]=AllocMem(BLOCKSIZE,MEMF_F
             AST);
352 5V           if (fastdiskpuffer[fastavail]==0)
353 clC             break;
354 iD6       }
355 JE3    }
356 MO     for (k=0 ; k<4 ; k++)
357 1K6       anzdest[k]=0;                    /*  reset amount of dest
          inations  */
358 Xo3    for (cylinder=flag[3] ; cylinder<=flag[4] ; cylinder=l) {

359 uh6       if (cylinder==flag[3]) {      /*  first time  */
360 GT9          if (flag[6]==flag[7]) {   /*  only one drive  */
361 YrC             if ((j=(int)(2*(flag[4]-flag[3])/(fastavail+avai
                l)+1.5))==1)
362 fRF                k=' ';
363 gTC             else
364 OVF                k='s';
365 WJC             locate(6,27);
366 kF              sprintf(string,"This copy will need %d swap%c.",
                j,k);
367 73              ConPutStr(WMsg,string);
368 OS              locate(7,18);ConPutStr(WMsg,"Please insert the \
                x9b0;33;40m"
369 NQX                                   "SOURCE\x9bm disk in drive
                                      \x9b0;33;40m");
370 9nC             sprintf(string,"df%d:\x9bm",flag[6]);ConPutStr(W
                Msg,string);
371 rE              locate(8,32);ConPutStr(WMsg,"and press any key")
                ;
372 3b              if ((j=Taste())==27)             /*  ESC  */
373 47F                return(FAILED);
374 GjC             do {                          /*  Is any disk inserted
                 ?  */
375 7cZ                                           /*  Abort if ESC is press
                                          ed  */
376 OOF                if ((j=ConMayGetChar(consoleReadPort,RMsg,&re
                   adstring))==27)
377 8BI                   return(FAILED);
378 8yF                diskreq[flag[6]]->iotd_Req.io_Command = TD_CH
                   ANGESTATE;
379 UV                 DoIO(diskreq[flag[6]]);
380 spC             } while(diskreq[flag[6]]->iotd_Req.io_Actual);
381 mG              locate(6,27) ; ConPutStr(WMsg,"
                                  ");
382 ng              locate(7,18);
383 yU              ConPutStr(WMsg,"
                                    ");
384 HO              locate(8,32);ConPutStr(WMsg,"                      ")
                ;
385 D19          }
386 zF           else {
387 PhC             locate (6,19);
388 ZS              ConPutStr(WMsg,"Please insert the \x9b0;33;40mSO
                URCE\x9bm disk "
389 IKR                            "in drive \x9b0;33;40m");
390 OsC             sprintf(string,"df%d:",flag[6]);ConPutStr(WMsg,s
                tring);
391 Hr              for (k=1 ; k<anzdisk ; k++)
392 3OF                if (flag[6+k]!=62 && flag[6+k]!='n') {
393 Y8I                   locate(6+k,23);sprintf(string,"\x9b0;33;40
                      m%d",k);
394 YU                    ConPutStr(WMsg,string);
395 p6                    ConPutStr(WMsg,"\x9b0;33;40m. DESTINATION\
                      x9bm disk in "
396 kGX                                  "drive \x9b0;33;40m");
397 YTI                   sprintf(string,"df%d:\x9bm",flag[6+k]);
398 cY                    ConPutStr(WMsg,string);
399 RwF                }
400 2gC             locate (6+k,32) ; ConPutStr(WMsg,"and press any
                key");
401 W4              if ((j=Taste())==27)            /*  ESC  */
402 XaF                return(FAILED);
403 SbC             do {            /*  Are disks in all drives ?  */

404 qSF                if ((j=ConMayGetChar(consoleReadPort,RMsg,&re
                   adstring))==27)
405 adI                   return(FAILED);
406 aQF                diskreq[flag[6]]->iotd_Req.io_Command = TD_CH
                   ANGESTATE;
407 wx                 DoIO(diskreq[flag[6]]);
408 N5C             }while (diskreq[flag[6]]->iotd_Req.io_Actual);
409 ux              for (k=1 ; k<anzdisk ; k++) {
410 LgF                if (flag[6+k]!=62 && flag[6+k]!='n') {
411 zpI                   do {
412 kLL                      if ((j=ConMayGetChar(consoleReadPort,RM
                         sg,&readstring))
413 tuR                            ==27)
414 JmO                         return(FAILED);
415 S2L                      diskreq[flag[6+k]]->iotd_Req.io_Command
                         =TD_CHANGESTATE;
416 kO                       DoIO(diskreq[flag[6+k]]);
417 olI                   } while(diskreq[flag[6+k]]->iotd_Req.io_Ac
                      tual);
418 kFF                }
419 lGC             }
420 wE              locate (6,19);
421 uR              ConPutStr(WMsg,"
                                ");
422 mM              for (k=1 ; k<anzdisk ; k++)
423 alF                if (flag[6+k] != 'n') {
424 o9I                   locate(6+k,23);
425 yV                    ConPutStr(WMsg,"
                                    ");
426 sNF                }
427 ggC             locate (6+k,32) ; ConPutStr(WMsg,"
                 ");
428 uP9          }
429 vQ6       }
430 c5        else                        /*  belongs to 'if (cylinder==f
          lag[3])'  */
431 oY9          if (flag[6]==flag[7])
432 qUC             if(WaitForDisk(flag[6],"SOURCE")==FAILED)
433 25F                return(FAILED);
434 QH6       MotorOn(flag[6]);
435 KK        for (i=0,l=cylinder ; i<avail-1 ; l++) {
436 hN9          if (l>flag[4])                        /*  end of copy
              ?  */
437 y7C             break;
438 8dO                        /*  Abort if ESC is pressed  */
439 ot9          if ((j=ConMayGetChar(consoleReadPort,RMsg,&readstri
             ng))==27) {
440 9dC             MotorOff(flag[6]);
441 AD              return(FAILED);
442 8d9          }
443 iu           if (flag[5]==2) {          /*  both surfaces  */
444 v6C             if (i==0 && fastavail>1) {
445 pWF                if((l=ReadFastTwoHeads(l))==FAILED)
446 FII                   return(FAILED);
447 zxF                i++;
448 EjC             }
449 Qf              if (l>flag[4] || i==avail-1)    /*  end of copy ?
                 */
450 BKF                break;
451 5gC             locate(6,30);sprintf(string,"Reading   cylinder
                %d ",l);
452 UQ              ConPutStr(WMsg,string);
453 N9              for (head=0 ; head<2 ; i++,head++) {
454 CAO reading1:
455 xeF                ReadCylSec(flag[6],l,head,i);
456 VZ                 if (diskreq[flag[6]]->iotd_Req.io_Error)
457 X5I                   if (diskreq[flag[6]]->iotd_Req.io_Error==2
                      9) {
458 1iL                      if(DiskRemoved()==FAILED)
459 SVO                         return(FAILED);
460 HJL                      goto reading1;
461 RwI                   }
462 H4                    else
```

Listing. »DCopy«, ein komfortables Kopierprogramm (Fortsetzung)

```
463 qkL                if((ReadError(i,l,head,
464 qMM                 diskreq[flag[6]]->iotd_Req.io_Error))=
                        =FAILED)
465 YbO                   return(FAILED);
466 W1C             }
467 X29          }
468 nh           else {                  /*  only one surface
             */
469 iPC             if (i==0 && fastavail) {  /*  Is there any fastm
                em ? */
470 KAF                if((l=ReadFastOneHead(l,flag[5]))==FAILED)
471 ehI                   return(FAILED);
472 OMF                i++;
473 d8C             }
474 aZ0 reading2:
475 q5C             if (l>flag[4] || i==avail-1)             /*  end
                 of copy ?  */
476 bkF                break;
477 V6C             locate(6,30);sprintf(string,"Reading   cylinder
                %d ",l);
478 uq              ConPutStr(WMsg,string);
479 nl              ReadCylSec(flag[6],l,flag[5],i);
480 rE              if (diskreq[flag[6]]->iotd_Req.io_Error)  /* err
                or detected */
481 vTF                if (diskreq[flag[6]]->iotd_Req.io_Error==29)
                   {
482 P6I                   if(DiskRemoved()==FAILED)
483 qtL                      return(FAILED);
484 jmI                   goto reading2;
485 pKF                }
486 fS                 else
487 E8I                   if((ReadError(i,l,head,
488 EkJ                    diskreq[flag[6]]->iotd_Req.io_Error))==FA
                        ILED)
489 wzL                      return(FAILED);
490 i1C             i++;        /*  increment memory-buffer  */
491 vQ9          }
492 wR6       }
493 ha        MotorOff(flag[6]);locate(6,30);ConPutStr(WMsg,"
                                  ");
494 aE        for (k=0 ; k<4 ; k++)
495 1D9          anz[k]=anzdest[k];      /*  remember the amount of
             destinations  */
496 wQ6       drive = flag[7];              /*  actual drive-number  */

497 fX        label8 = label9 = 0;          /*  reset multiple-copy lab
             els  */
498 9L        firsttime = YES;              /*  flag  */
499 X20 change:
500 dS6       if (anzdest[drive] || flag[6]==drive){
501 4u9          if (flag[6]==drive || anz[drive]<anzdest[drive] ||
             changeflag==YES){
502 GmC             changeflag = NO;
503 BU              if (firsttime==YES)
504 xuE                sprintf(meldung,"%d. DESTINATION",anz[drive]+1
                   );
505 ylC             else
506 rtE                sprintf(meldung,"%d. DESTINATION",anzdest[driv
                   e]-anz[drive]+1);
507 aMC             if(WaitForDisk(drive,meldung)==FAILED)
508 FIF                return(FAILED);
509 k7C             MotorOn(drive);
510 EJ9          }
511 Fk6       }
512 5s        else
513 oB9          MotorOn(drive);
514 V7V                                     /* writing data */
515 cc6       for (i=0,l=cylinder ; i<avail-1 ; l++) {
516 zf9          if (l>flag[4])              /*  end of copy ?  */
517 GPC             break;
518 KLV                                     /*  ESC pressed ?  */
519 6B9          if ((j=ConMayGetChar(consoleReadPort,RMsg,&readstri
             ng))==27) {
520 41C             MotorOff(drive);
521 SV              return(FAILED);
522 Qv9          }
523 OC           if (flag[5]==2) {          /*  both surfaces  */
524 3rC             if (i==0 && fastavail>1) {   /*  writing fastme
                m-buffer  */
525 6PF                if((l=WriteFastTwoHeads(l,drive))==FAILED)
526 XaI                   return(FAILED);
527 2DF                else if (l==NO)
528 TaI                   goto giveup;
529 JHF                i++;
530 Y3C             }
531 jX              if (l>flag[4] || i==avail-1)
532 VeF                break;
533 m9C             verify=0;              /*  reset amount of verify
                -errors  */
534 7j              locate(6,30);
535 d2              sprintf(string,"Writing   cylinder %d ",l);
536 qm              ConPutStr(WMsg,string);
537 jV              for (head=0 ; head<2 ; i++,head++) {
538 DsO writing2:
539 a4F                if(WriteCylSec(drive,l,head,i)==FAILED)
540 loI                   return(FAILED);
541 InF                if (flag[0]=='y') {         /*  verify ON  */
542 jOI                   ReadCylSec(drive,l,head,i);
543 wY                    if (diskreq[drive]->iotd_Req.io_Error)
544 uAL                      if (diskreq[drive]->iotd_Req.io_Error==
                         29) {
545 GyO                         locate(6,22);
546 HP                          ConPutStr(WMsg,"Destination is \x9b0
                            ;33;40m"
547 U4d                                       "removed\x9bm. Try it
                                          again !");
548 WTO                         MotorOff(drive);
549 VN                          if(WaitForDisk(drive,"")==FAILED)
550 vyR                            return(FAILED);
551 SDO                         locate(6,25);
552 86                          ConPutStr(WMsg,"
                                          ");
553 Sp                          MotorOn(drive);
554 SB                          goto writing2;
555 xSL                      }
556 6s                       else {          /*  verify-error detecte
                         d  */
557 MfO                         if (++verify < flag[1]) {
558 4UR                            if((VerifyError(l,head,drive))==F
                               AILED)
559 47U                               return(FAILED);
560 reR                            else
561 ZIU                               goto writing2;
562 4ZO                         }
563 q6                          else {
564 m6R                            anzdest[drive]--;
565 4B                             goto giveup;
566 8dO                         }
567 9eL                      }
568 AfF                }
569 BgC             }
570 Ch9          }
571 Yi           else {                         /*  only one surface  *
             /
572 OUC             if (i==0 && fastavail) { /*  copying from fastme
                m to chipmem  */
573 4XF                if((l=WriteFastOneHead(l,drive))==FAILED)
574 JMI                   return(FAILED);
575 ozF                else if (l==NO)
576 FMI                   goto giveup;
577 53F                i++;
578 KpC             }
579 OhO writing4:
580 WKC             if (l>flag[4] || i==avail-1)
581 IRF                break;
582 y5C             if(WriteCylSec(drive,l,flag[5],i)==FAILED)
583 SVF                return(FAILED);
584 3qC             if (flag[0]=='y') {        /*  verify  */
585 PeF                ReadCylSec(drive,l,flag[5],i);
586 dF                 if (diskreq[drive]->iotd_Req.io_Error)
587 brI                   if (diskreq[drive]->iotd_Req.io_Error==29)
                      {
588 xfL                      locate(6,22);
589 y6                       ConPutStr(WMsg,"Destination is \x9b0;33
                         ;40m"
590 Bla                                    "removed\x9bm. Try it ag
                                       ain !");
591 DAL                      MotorOff(drive);
592 C4                       if(WaitForDisk(drive,"")==FAILED)
593 8t                       locate(6,25);
594 om                       ConPutStr(WMsg,"
                                       ");
595 8V                       MotorOn(drive);
596 G1                       goto writing4;
597 d8I                   }
598 Pf                    else {
599 2LL                      if (++verify < flag[1]) {
600 8rO                         if((VerifyError(l,flag[5],drive))==F
                            AILED)
601 knR                            return(FAILED);
602 XKO                         else
603 N8R                            goto writing4;
604 kFL                      }
605 Wm                       else {
606 SmO                         anzdest[drive]--;
607 kr                          goto giveup;
608 oJL                      }
609 pKI                }
610 qLC          }
611 db           i++;
```

```
612 sN9          }
613 t06        }
614 ux0 giveup:
615 XA6        MotorOff(drive);locate(6,30);ConPutStr(WMsg,"
                  ");
616 CL         if (flag[2]=='y') {
617 e49           if (cylinder==flag[3]) {
618 nVC              locate (6,27) ; ConPutStr(WMsg,"Another DESTINAT
                     ION (y/n) ?");
619 JV               j=Taste();
620 Jc               locate (6,27) ; ConPutStr(WMsg,"
                            ");
621 z0               if (j=='y' || j=='Y') {
622 IdF                 anz[drive]=++anzdest[drive];
623 DN                  changeflag = YES;
624 V4                  goto change;
625 5aC              }
626 6b9           }
627 vW            else if (anz[drive]-- > 0) {
628 7mC              firsttime = NO;
629 a9               goto change;
630 Af9           }
631 Bg6        }
632 Ne         if (flag[8]!='n' && flag[8]!=62 && label8==0) {
633 Dn9           drive=flag[8];
634 SZ            label8=1;
635 gF            goto change;
636 G16        }
637 cr         if (flag[9]!='n' && flag[9]!=62 && label9==0) {
638 Lw9           drive=flag[9];
639 Zh            label9=1;
640 1K            goto change;
641 Lq6        }
642 XI3     if (verify==flag[1])
643 QT6        return(FAILED);
644 Ot3     }
645 kJ      return(OK);
646 1q0 }    /*  end of Copy()  */
647 9B0                        /*  exit gracefully  */
648 KpO End()
649 Ps  {
650 yU3     int i;
651 1g      if (flag[6]!=flag[7]) {     /*  more than one drive  */
652 AE6        ConPutStr(WMsg,"\f");
653 we         locate(4,24);
654 uY         ConPutStr(WMsg,"Please remove ALL destinations,");
655 8u         locate(5,27);
656 Jz         ConPutStr(WMsg,"so DOS won't get confused.");
657 Gw         locate(7,33);
658 m5         ConPutStr(WMsg,"Press any key.");
659 v6         i=Taste();
660 e93     }
661 zX      for (i=0 ; i<4 ; i++)
662 oh6        if (i!=t[0]-'0' && i!=t[1]-'0' && i!=t[2]-'0' && i!=t[
           3]-'0') {
663 S09           CloseDevice(diskreq[i]);
664 sc            if((mp = FindTask("File System"))!=NULL)
665 yfA            dos_packet(&mp->pr_MsgPort,ACTION_INHIBIT,FALSE,OL
               ,OL,OL,OL,OL,OL);
666 FX9           DeleteExtIO(diskreq[i],(long)sizeof(struct IOExtTD)
              );
667 hS            DeletePort(diskport[i]);
668 mH6        }
669 a63     for (i=0;i<avail;i++)
670 p26        FreeMem(diskpuffer[i],BLOCKSIZE);
671 sP3     for (i=0;i<fastavail;i++)
672 Fu6        FreeMem(fastdiskpuffer[i],BLOCKSIZE);
673 yx3     i=CheckIO(RMsg);
674 3a      if (i==FALSE)
675 Me6        AbortIO(RMsg);
676 WA3     WaitPort(consoleReadPort);
677 43      GetMsg(consoleReadPort);
678 v2      CloseDevice(WMsg);
679 qS      CloseWindow(w);
680 t2      DeleteStdIO(RMsg);
681 SK      DeletePort(consoleReadPort);
682 FT      DeleteStdIO(WMsg);
683 KB      DeletePort(consoleWritePort);
684 Js      CloseLibrary(IntuitionBase);
685 En      exit (OL);
686 z70 }   /*  end of End()  */
687 JY  Changenum(i)
688 a6  int i;
689 3W  {
690 o23     diskreq[i]->iotd_Req.io_Command = TD_CHANGENUM;
691 kE      DoIO(diskreq[i]);
692 AfO }
693 kp  MotorOn(i)
694 gC  int i;
695 9c  {
696 7s3     diskreq[i]->iotd_Req.io_Length = 1;
697 I7      diskreq[i]->iotd_Req.io_Command = TD_MOTOR;
698 rL      DoIO(diskreq[i]);
699 Hm0 }
700 1d  MotorOff(i)
701 nJ  int i;
702 GJ  {
703 9t3     diskreq[i]->iotd_Req.io_Length = 0;
704 PE      diskreq[i]->iotd_Req.io_Command = TD_MOTOR;
705 yS      DoIO(diskreq[i]);
706 0t0 }
707 w6  ReadCylSec(i,cyl,hd,puff)
708 uQ  int i;
709 k5  SHORT cyl,hd;
710 ZG  int puff;
711 Ps  {
712 8B3     LONG offset;
713 sy      diskreq[i]->iotd_Req.io_Length = BLOCKSIZE;
714 UQ      diskreq[i]->iotd_Req.io_Data = (APTR)diskpuffer[puff];
715 kE      diskreq[i]->iotd_Req.io_Command = CMD_READ;
716 6S      offset = TD_SECTOR * (NUMSECS * hd + NUMSECS * NUMHEADS *
         cyl);
717 OR      diskreq[i]->iotd_Req.io_Offset = offset;
718 Bf      DoIO(diskreq[i]);
719 tx      return(YES);
720 c70 }
721 Xv  WriteCylSec(i,cyl,hd,puff)
722 8e  int i;
723 yJ  SHORT cyl,hd;
724 nU  int puff;
725 d6  {
726 MP3     LONG offset;
727 LJ0 label:
728 7D3     diskreq[i]->iotd_Req.io_Length = BLOCKSIZE;
729 Jf      diskreq[i]->iotd_Req.io_Data = (APTR)diskpuffer[puff];
730 5S      diskreq[i]->iotd_Req.io_Command = TD_FORMAT;
731 Lh      offset = TD_SECTOR * (NUMSECS * hd + NUMSECS * NUMHEADS *
         cyl);
732 dg      diskreq[i]->iotd_Req.io_Offset = offset;
733 Qu      DoIO(diskreq[i]);
734 zl      if (diskreq[i]->iotd_Req.io_Error) {
735 Sm6        if (diskreq[i]->iotd_Req.io_Error == 28) {
736 RJ9           locate(6,18);
737 hA            ConPutStr(WMsg,"Destination is \x9b0;33;40mwrite-pr
              otected\x9bm. "
738 U90                          "Try it again !");
739 e09           MotorOff(i);
740 k7            if(WaitForDisk(i,"")==FAILED)
741 03C              return(FAILED);
742 XP9           locate(6,18);
743 mI            ConPutStr(WMsg,"
                         ");
744 Rx            MotorOn(i);
745 Zb            goto label;
746 2X6        }
747 J4         if (diskreq[i]->iotd_Req.io_Error == 29) {
748 XF9           locate(6,22);
749 qa            ConPutStr(WMsg,"Destination is \x9b0;33;40mremoved\
              x9bm. "
750 gL0                          "Try it again !");
751 qC9           MotorOff(i);
752 wJ            if(WaitForDisk(i,"")==FAILED)
753 CFC              return(FAILED);
754 dL9           locate(6,22);
755 Ip            ConPutStr(WMsg,"
                 ");
756 d9            MotorOn(i);
757 ln            goto label;
758 Ej6        }
759 Fk3     }
760 Yc      return(YES);
761 Hm0 }
762 20  FullSeek(times,unit)
763 zs  int times,unit;
764 Gj  {
765 pL3     int i;
766 tw      for (i=0 ; i<times ; i++) {     /*  Move head to cyl 79 th
         en to cyl 0  */
767 T16        diskreq[unit]->iotd_Req.io_Offset=
768 eTV                          (long)(((NUMCYLS-1)*NUMSECS*N
                             UMHEADS-1)*512);
769 hu6        diskreq[unit]->iotd_Req.io_Command = TD_SEEK;
770 Je         DoIO(diskreq[unit]);
771 oT         diskreq[unit]->iotd_Req.io_Offset=0L;
772 kx         diskreq[unit]->iotd_Req.io_Command = TD_SEEK;
773 mh         DoIO(diskreq[unit]);
774 Uz3     }
775 V00 }
776 DS  WaitForDisk(c,text)
777 RL  int c;
```

Listing. »DCopy« ein komfortables Kopierprogramm (Fortsetzung)

```
778 wc  char *text;
779 Vy  {
780 Ac3     int j;
781 rf      locate(7,(80-38-strlen(text))/2);
782 z4      ConPutStr(WMsg,"Please insert the \x9b0;33;40m");
783 8H      ConPutStr(WMsg,text);
784 3n      sprintf(string,"\x9bm disk in drive \x9b0;33;40mdf%d: \x9
        bm",c);
785 rn      ConPutStr(WMsg,string);
786 ma      do {            /*  Is disk removed  */
787 wJ6        if((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))
        ==27) {
788 1o9           return(FAILED);
789 JE6        }
790 d9         diskreq[c]->iotd_Req.io_Command = TD_CHANGESTATE;
791 sG         DoIO(diskreq[c]);
792 4D3     }while(!diskreq[c]->iotd_Req.io_Actual);
793 Ez      do {            /*  Is disk inserted ?  */
794 U06        j=ConMayGetChar(consoleReadPort,RMsg,&readstring);
795 bH         if (j==27)
796 tw9           return(FAILED);
797 kG6        diskreq[c]->iotd_Req.io_Command = TD_CHANGESTATE;
798 zN         DoIO(diskreq[c]);
799 Ph3     }while(diskreq[c]->iotd_Req.io_Actual);
800 H2      locate(7,10);
801 JQ      ConPutStr(WMsg,"
                    "
802 FtI                  "      ");
803 FJ3     return(YES);
804 yT0 }
805 Ji  Taste()
806 wP  {
807 b33     int j;
808 uF      while((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))
        !=-1);
809 UF      while((j=ConGetChar(consoleReadPort,RMsg,&readstring))=='
        \n');
810 H8      return(j);
811 5a0 }
812 Po  locate(lin,col)                 /*  analog BASIC-locate  */
813 Jt  int lin,col;
814 4X  {
815 Cv3     sprintf (string,"\x9b%d;%dH",lin,col);
816 MI      ConPutStr(WMsg,string);
817 Bg0 }
818 6D  ReadFastTwoHeads(l)
819 zJ  int l;
820 Ad  {
821 MG3     int j,k,head;
822 X6      for (k=0 ; k<fastavail-1 ; l++) {
823 vA6        if (l>flag[4])
824 dW9           return(l);
825 YL6        if((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))
        ==27) {
826 Nr9           MotorOff(flag[6]);
827 OR            return(FAILED);
828 Mr6        }
829 sU         locate(6,30);
830 78         sprintf(string,"Reading   cylinder %d ",l);
831 bX         ConPutStr(WMsg,string);
832 kd         for (head=0 ; head<2 ; head++) {
833 RR0 reading3:
834 J59           ReadCylSec(flag[6],l,head,0);
835 cg            if (diskreq[flag[6]]->iotd_Req.io_Error)
836 eCC              if (diskreq[flag[6]]->iotd_Req.io_Error==29) {
837 8pF                 if(DiskRemoved()==FAILED)
838 ZcI                    return(FAILED);
839 WaF                 goto reading3;
840 Y3C              }
841 OB               else
842 R1F                 if((ReadError(0,l,head,
843 xTG                  diskreq[flag[6]]->iotd_Req.io_Error))==FAILE
                     D)
844 fiI                    return(FAILED);
845 Gg9           erase=fastdiskpuffer[k++] ; puffmerk=diskpuffer[0];

846 S80 #asm
847 RLB                movem.l   d0/a0-a1,-(sp)
848 5D                 move.w    #1407,d0
849 LU                 move.l    _erase,a0
850 BK                 move.l    _puffmerk,a1
851 g30 laber            move.l    (a1)+,(a0)+
852 47D                  dbra      d0,laber
853 uzB                movem.l   (sp)+,d0/a0-a1
854 QK0 #endasm
855 nI6        }
856 oJ3     }
857 A3      return(l);
858 qL0 }
859 dn  ReadFastOneHead(l,head)
860 nF  int l,head;
861 pI  {
862 NE3     int j,k;
863 db      for (k=0 ; k<fastavail ; l++) {
864 ap6        if (l>flag[4])
865 IB9           return(l);
866 T56        locate(6,30);
867 iJ         sprintf(string,"Reading   cylinder %d ",l);
868 C8         ConPutStr(WMsg,string);
869 560 reading4:
870 H46        if((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))
        ==27) {
871 6a9           MotorOff(flag[6]);
872 7A            return(FAILED);
873 5a6        }
874 xj         ReadCylSec(flag[6],l,head,0);
875 GK         if (diskreq[flag[6]]->iotd_Req.io_Error)
876 Iq9           if (diskreq[flag[6]]->iotd_Req.io_Error==29) {
877 mTC              if(DiskRemoved()==FAILED)
878 DGF                 return(FAILED);
879 EJC              goto reading4;
880 Ch9           }
881 2p            else
882 5fC              if((ReadError(0,l,head,
883 b7D               diskreq[flag[6]]->iotd_Req.io_Error))==FAILED)
884 JMF                 return(FAILED);
885 uK6        erase=fastdiskpuffer[k++] ; puffmerk=diskpuffer[0];
886 6m0 #asm
887 5z8           movem.l   d0/a0-a1,-(sp)
888 Jr            move.w    #1407,d0
889 z8            move.l    _erase,a0
890 py            move.l    _puffmerk,a1
891 1V0 laber0      move.l    (a1)+,(a0)+
892 kYA             dbra      d0,laber0
893 Yd8           movem.l   (sp)+,d0/a0-a1
894 4y0 #endasm
895 Rw3     }
896 ng      return(l);
897 Ty0 }
898 2A  ReadError(i,track,head,errnumber)
899 PF  int i,track,head,errnumber;
900 Sv  {
901 7Z3     int j;
902 PT      locate(10,1);
903 95      ConPutStr(WMsg,"\x9b0;33;40mRead-error\x9bm at cylinder =
        ");
904 Ra      sprintf(string,"%d, head = %d error = %d",track,head,errn
        umber);
905 nJ      ConPutStr(WMsg,string);
906 LQ      if ((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))==
        27) {
907 gA6        MotorOff(flag[6]);
908 hk         return(FAILED);
909 fA3     }
910 bN      erase=diskpuffer[i];
911 FN      for (j=0;j<BLOCKSIZE;j++)
912 GT6        *erase++ = '\0';
913 6V3     locate(10,60);
914 qx      sprintf(string,"Read-errors: %d",++error);
915 xt      ConPutStr(WMsg,string);
916 7g      return(OK);
917 nI0 }
918 Za  DiskRemoved()
919 1E  {
920 N73     locate(6,24);
921 zo      ConPutStr(WMsg,"Source is \x9b0;33;40mremoved\x9bm. Try i
        t again !");
922 vP      MotorOff(flag[6]);
923 iT      if(WaitForDisk(flag[6],"")==FAILED)
924 x06        return(FAILED);
925 SC3     locate(6,24);
926 3a      ConPutStr(WMsg,"                                    ");
927 NE      MotorOn(flag[6]);
928 GK      return(YES);
929 zU0 }
930 S3  OpenConsole(writerequest,readrequest,window)
931 RQ  struct IOStdReq *writerequest;
932 xe  struct IOStdReq *readrequest;
933 Ka  struct Window *window;
934 OT  {
935 F53     int error;
936 o0      writerequest->io_Data = (APTR) window;
937 Sq      writerequest->io_Length = sizeof(*window);
938 do      error = OpenDevice("console.device", OL, writerequest, OL
        );
939 c1      readrequest->io_Device = writerequest->io_Device;
940 ur      readrequest->io_Unit   = writerequest->io_Unit;
941 UB      /* clone required parts of the request */
942 Z9      return(error);
943 D10 }
```

```
944 Wp  /* Output a single character to a specified console */
945 e8  ConPutChar(request,character)
946 Lq  struct IOStdReq *request;
947 NX  char character;
948 Eh  {
949 0d3     request->io_Command = CMD_WRITE;
950 CF      request->io_Data = (APTR)&character;
951 Ly      request->io_Length = 1L;
952 nU      DoIO(request);
953 fJ      return(YES);
954 Ot0 }
955 Ms  ConPutStr(request,string)
956 V0  struct IOStdReq *request;
957 1w  char *string;
958 Or  {
959 An3     request->io_Command = CMD_WRITE;
960 2z      request->io_Data = (APTR)string;
961 hD      request->io_Length = -1L;  /* tells console to end when i
        t
962 zWV                                * sees a terminating zero on
963 Bw                                 * the string. */
964 zg3     DoIO(request);
965 rv      return(YES);
966 a50 }
967 HH  QueueRead(request,whereto)
968 hC  struct IOStdReq *request;
969 Kf  char *whereto;
970 a3  {
971 S63     request->io_Command = CMD_READ;
972 TI      request->io_Data = (APTR)whereto;
973 hK      request->io_Length = 1L;
974 jV      SendIO(request);
975 15      return(YES);
976 kF0 }
977 x3  ConMayGetChar(consolePort,request,whereto)
978 9S  struct Port *consolePort;
979 sN  struct IOStdReq *request;
980 Vq  char *whereto;
981 1E  {
982 5w3     register temp;
983 cU      if ( GetMsg(consolePort) == NULL ) return(-1);
984 qT6         temp = *whereto;
985 sJ3     QueueRead(request,whereto);
986 Cp      return(temp);
987 vQ0 }
988 iQ  ConGetChar(consolePort,request,whereto)
989 2X  struct IOStdReq *request;
990 a2  struct MsgPort *consolePort;
991 g1  char *whereto;
992 wP  {
993 FB3     register UBYTE temp;
994 4c      while((GetMsg(consolePort) == NULL)) WaitPort(consolePort
        );
995 QP      temp = *whereto;   /* get the character */
996 3U      QueueRead(request,whereto);
997 NO      return(temp);
998 6b0 }
999 Sq  CursorOn()
1000 4X {
1001 nP3    ConPutStr(WMsg,"\x9b\x20\x70");
1002 Af0 }
1003 Cf  CursorOff()
1004 8b  {
1005 783    ConPutStr(WMsg,"\x9b\x30\x20\x70");
1006 EjO }
1007 HB  long
1008 IO  dos_packet(port,type,arg1,arg2,arg3,arg4,arg5,arg6,arg7)
1009 Ar  struct MsgPort *port;
1010 xK  long type, arg1, arg2, arg3, arg4, arg5, arg6, arg7;
1011 Fi  {
1012 5N3    register struct StandardPacket *sp;
1013 xy     register struct MsgPort *rp;
1014 Ta     char *zz;
1015 z7     long ret;
1016 Or     if ((rp = CreatePort(0L, 0L)) == 0)
1017 SV6        return(FAILED);
1018 F83    if ((sp = (struct StandardPacket *)
1019 U9D             AllocMem((long)sizeof(*sp), MEMF_PUBLIC| MEMF_CL
                EAR)) == 0) {
1020 fU6        DeletePort(rp);
1021 WZ         return(FAILED);
1022 Uz3    }
1023 pX     zz="BBBBB";
1024 eN     sp->sp_Msg.mn_Node.ln_Name = (char *)&sp->sp_Pkt;
1025 5W     sp->sp_Pkt.dp_Link = &sp->sp_Msg;
1026 yT     sp->sp_Pkt.dp_Port = rp;
1027 8H     sp->sp_Pkt.dp_Type = type;
1028 AE     sp->sp_Pkt.dp_Arg1 = arg1;
1029 LM     sp->sp_Pkt.dp_Arg2 = arg2;
1030 WU     sp->sp_Pkt.dp_Arg3 = arg3;
1031 hc     sp->sp_Pkt.dp_Arg4 = arg4;
1032 sk     sp->sp_Pkt.dp_Arg5 = arg5;
1033 3s     sp->sp_Pkt.dp_Arg6 = arg6;
1034 E0     sp->sp_Pkt.dp_Arg7 = arg7;
1035 mn     if (arg1 == TRUE) {
1036 7n6        locate(6,20);
1037 Ip         ConPutStr(WMsg,"DOS is confused. Please reset your Ami
           ga.");
1038 kF3    }
1039 41     PutMsg(port, &sp->sp_Msg);
1040 Br     locate(6,20);
1041 a6     ConPutStr(WMsg,"
       ");
1042 8A     WaitPort(rp);
1043 fs     GetMsg(rp);
1044 Xd     ret = sp->sp_Pkt.dp_Res1;
1045 1o     FreeMem(sp, (long)sizeof(*sp));
1046 5u     DeletePort(rp);
1047 ZQ     return(ret);
1048 uP0 }
1049 DR  WriteFastTwoHeads(l,drive)
1050 YE  int l,drive;
1051 tM  {
1052 QG3    int head;
1053 SJ     int j,k;
1054 QW     int verify=0;            /*  reset amount of verify-erro
       rs  */
1055 Ir     for (k=0 ; k<fastavail-1 ; l++) {
1056 gv6        if (l>flag[4])
1057 OH9            return(l);
1058 J66        if((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))
           ==27) {
1059 1i9            MotorOff(drive);
1060 9C             return(FAILED);
1061 7c6        }
1062 dF         locate(6,30);
1063 9Y         sprintf(string,"Writing   cylinder %d ",l);
1064 MI         ConPutStr(WMsg,string);
1065 V0         for (head=0 ; head<2 ; head++) {
1066 pF9            erase=fastdiskpuffer[k++] ; puffmerk=diskpuffer[0];

1067 1h0 #asm
1068 Ou9            movem.l    d0/a0-a1,-(sp)
1069 em             move.w    #1407,d0
1070 u3             move.l    _erase,a0
1071 kt             move.l    _puffmerk,a1
1072 c60 laber3       move.l    (a0)+,(a1)+
1073 rDB              dbra      d0,laber3
1074 TY9            movem.l    (sp)+,d0/a0-a1
1075 zt0 #endasm
1076 pT  writing1:
1077 fE9            if(WriteCylSec(drive,l,head,0)==FAILED)
1078 RUC                return(FAILED);
1079 yT9            if (flag[0]=='y') {        /*  verify ON  */
1080 tEC                ReadCylSec(drive,l,head,0);
1081 cE                 if (diskreq[drive]->iotd_Req.io_Error)
1082 aqF                    if (diskreq[drive]->iotd_Req.io_Error==29) {
1083 weI                        locate(6,22);
1084 x5                         ConPutStr(WMsg,"Destination is \x9b0;33;40
                           m"
1085 AkX                                   "removed\x9bm. Try it again
                                   !");
1086 C9I                        MotorOff(drive);
1087 B3                         if(WaitForDisk(drive,"")==FAILED)
1088 beL                            return(FAILED);
1089 8tI                        locate(6,25);
1090 om                         ConPutStr(WMsg,"
                                   ");
1091 8V                         MotorOn(drive);
1092 4m                         goto writing1;
1093 d8F                    }
1094 mY                     else {        /*  verify-error detected  */
1095 2LI                        if (++verify < flag[1]) {
1096 kAL                            if((VerifyError(l,head,drive))==FAILED)
1097 kn0                                return(FAILED);
1098 XKL                            else
1099 Bt0                                goto writing1;
1100 kFI                        }
1101 Wm                         else {
1102 SmL                            anzdest[drive]--;
1103 00                             return(NO);
1104 oJI                        }
1105 pKF                    }
1106 qL9                }
1107 rM6            }
1108 sN3        }
1109 E7         return(l);
1110 uP0 }
1111 V1  VerifyError(l,head,drive)
1112 hV  int l,head,drive;
```

Listing. »DCopy« ein komfortables Kopierprogramm (Fortsetzung)

```
1113 tM  {
1114 Y03     int j;
1115 ny      locate (6,21);
1116 35      ConPutStr(WMsg,"\x9b0;33;40m"
1117 7HI                     "Verify Error\x9bm at cylinder ");
1118 is3     sprintf(string,"%d, Head %d !!!",l,head);
1119 FB      ConPutStr(WMsg,string);
1120 8d0                               /*  Abort if ESC is pressed  */
1121 ot3     if ((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))==
             27) {
1122 mj6        MotorOff(drive);
1123 AD         return(FAILED);
1124 8d3     }
1125 Re      FullSeek(2,drive);    /*  move head one time  */
1126 y9      locate (6,21);
1127 kR      ConPutStr(WMsg,"                                        ")
             ;
1128 X6      return(OK);
1129 Di0 }
1130 ft  WriteFastOneHead(l,drive)
1131 rX  int l,drive;
1132 Cf  {
1133 kb3     int j,k;
1134 mB      int verify=0;
1135 wb      for (k=0 ; k<fastavail ;) {
1136 yD6        if (l>flag[4])
1137 gZ9           return(l);
1138 b06        if((j=ConMayGetChar(consoleReadPort,RMsg,&readstring))
                ==27) {
1139 309           MotorOff(drive);
1140 RU            return(FAILED);
1141 Pu6        }
1142 3T         erase=fastdiskpuffer[k++] ; puffmerk=diskpuffer[0];
1143 Fv0 #asm
1144 E86        movem.l   d0/a0-a1,-(sp)
1145 s0         move.w    #1407,d0
1146 8H         move.l    _erase,a0
1147 y7         move.l    _puffmerk,a1
1148 JM0 laber4     move.l   (a0)+,(a1)+
1149 9WA            dbra     d0,laber4
1150 hm6        movem.l   (sp)+,d0/a0-a1
1151 D70 #endasm
1152 Br  writing3:
1153 Qc6        if(WriteCylSec(drive,l,flag[5],0)==FAILED)
1154 fi9           return(FAILED);
1155 G36        if (flag[0]=='y') {        /*  verify  */
1156 wG9           ReadCylSec(drive,l,flag[5],0);
1157 qS            if (diskreq[drive]->iotd_Req.io_Error)
1158 o4C              if (diskreq[drive]->iotd_Req.io_Error==29) {
1159 AsF                 locate(6,22);
1160 BJ                  ConPutStr(WMsg,"Destination is \x9b0;33;40m"
1161 OyU                                "removed\x9bm. Try it again !"
                                        );
1162 QNF                 MotorOff(drive);
1163 PH                  if(WaitForDisk(drive,"")==FAILED)
1164 psI                    return(FAILED);
1165 M7F                 locate(6,25);
1166 20                  ConPutStr(WMsg,"
                               ");
1167 Mj                  MotorOn(drive);
1168 QA                  goto writing3;
1169 rMC              }
1170 dt               else {
1171 GZF                 if (++verify < flag[1]) {
1172 M5I                    if((VerifyError(l,flag[5],drive))==FAILED)
1173 y1L                       return(FAILED);
1174 lYI                    else
1175 XHK                      goto writing3;
1176 yTF                 }
1177 k0                  else {
1178 g0I                    anzdest[drive]--;
1179 cE                     return(NO);
1180 2XF                 }
1181 3YC              }
1182 4Z6        }
1183 xy         l++;
1184 6b3     }
1185 SL      return(l);
1186 8d0 }
(C) 1987 M&T
```

Listing. »DCopy«, ein komfortables Kopierprogramm (Schluß)

Achtung:

Wir machen unsere Inserenten darauf aufmerksam, daß das **Angebot**, der **Verkauf** oder die **Verbreitung** von urheberrechtlich geschützter Software nur für Originalprogramme erlaubt ist.

Das Herstellen, Anbieten, Verkaufen und Verbreiten von **»Raubkopien«** verstößt gegen das Urheberrechtsgesetz und kann straf- und zivilrechtlich verfolgt werden. Bei Verstößen muß mit Anwalts- und Gerichtskosten von über DM 1 000,— gerechnet werden.

Originalprogramme sind am Copyright-Hinweis und am Originalaufkleber des Datenträgers (Diskette oder Kassette) zu erkennen und normalerweise originalverpackt. Mit dem Kauf von Raubkopien erwirbt der Käufer auch kein Eigentum und geht das Risiko einer jederzeitigen Beschlagnahmung ein.

Wir bitten unsere Leser in deren eigenem Interesse, Raubkopien von Original-Software weder anzubieten, zu verkaufen noch zu verbreiten. Erziehungsberechtigte haften für ihre Kinder.

Der Verlag wird in Zukunft keine Anzeigen mehr veröffentlichen, die darauf schließen lassen, daß Raubkopien angeboten werden.

CLI – die zweite Haut des Amiga

Batch-Dateien erlauben dem Anwender, durch einen einzigen Aufruf viele Befehle der Reihe nach abzuarbeiten. In diesem Kursteil erfahren Sie, was alles mit einer Batch-Datei gemacht werden kann.

Die letzte Folge behandelte Befehle, die sich zum Einbau in eine Batch-Datei (Kommandodatei, Stapeldatei) eignen. Batch-Dateien sind kleine Programme, die mehrere CLI-Befehle enthalten können. Beim Aufruf dieser Datei werden die einzelnen Befehle der Reihe nach abgearbeitet. Dieses Thema wird nun weitergeführt mit einem Beispiel, wie eine Stapeldatei aussehen könnte.

Um eine flexible Ausführungsdatei programmieren zu können, stellt das CLI verschiedene Befehle zur Verfügung wie etwa IF..THEN..ELSE ..ENDIF-Konstruktionen, die im letzten Teil besprochen wurden. Es existiert noch eine zweite Befehlskombination, die in etwa mit dem von Basic her bekannten Befehl GOTO gleichzusetzen ist. Es sind dies die Befehle SKIP und LAB.

LAB dient dazu, eine Sprungmarke zu fixieren, zu der mit SKIP gesprungen werden kann. Ein Beispiel:

```
BINDDRIVERS
LOADWB
...
...
SKIP Marke1
...
...
LAB Marke1
...
```

Alle Befehle, die auf die LAB-Zeile folgen, werden nach dem Sprung zu der definierten Marke weiterbearbeitet. Es müssen nur noch die Bedingungen definiert werden, bei denen ein Sprung erfolgen soll (siehe Schema in Bild 1). Eine Möglichkeit dafür ist eine IF-Abfrage:

```
...
...
IF EXISTS Preferences
 THEN setze Abarbeitung
fort
 ELSE SKIP Marke1
ENDIF
...
...
ENDCLI > NIL:
LAB Marke1
ECHO "Datei nicht vorhanden"
ENDCLI > NIL:
```

Bei unserem Beispiel wird zuerst geprüft, ob die Datei »Preferences« auf dem aktuellen Laufwerk vorhanden ist. Wenn ja, arbeitet der Amiga die Batch-Datei weiter ab. Ist die Datei aber nicht vorhanden, springt das Betriebssystem mit SKIP zu der mit LAB definierten Marke und führt die Befehle aus, die der LAB-Zeile folgen. Anschließend wird das CLI-Fenster geschlossen. Diese Kombination kann also eingesetzt werden, um mehr Struktur in eine Stapeldatei zu bringen. Allerdings ist es wohl in den seltensten Fällen nötig, diese Kombination zu verwenden, da sich ein solches Problem auch anders lösen läßt.

Ein weiterer, allerdings selten gebrauchter Befehl ist QUIT. Dieser Befehl dient zum Beenden von Stapeldateien und Kommandofolgen und kann beispielsweise anstelle von ENDCLI stehen. Im Unterschied zu ENDCLI wird bei QUIT das CLI-Fenster nicht geschlossen. Verändern wir doch das obige Beispiel, indem wir QUIT einbauen:

```
...
...
IF EXISTS Preferences
 THEN setze Abarbeitung
fort
 ELSE SKIP Marke1
ENDIF
...
...
QUIT
LAB Marke1
ECHO "Datei nicht vorhanden"
QUIT
```

Die Datei tut nichts anderes als oben beschrieben. Einziger Unterschied ist, daß nach der Abarbeitung das CLI-Fenster nicht geschlossen wird, was bei ENDCLI der Fall wäre.

Quit kann aber noch mehr: Wenn hinter dem Befehl ein Parameter angegeben wird, setzt Amiga-DOS diesen Wert nach Beendigung des Befehls automatisch als neues Fehlerniveau fest:

```
QUIT 40
```

Nach Beendigung des QUIT-Befehls würde das Fehlerniveau also den Wert 40 aufweisen. Der Befehl, der wie auch SKIP und LAB nur in Stapeldateien verwendet werden kann, ist sicher nicht dafür geeignet, dauernd eingesetzt zu werden. In speziellen Situationen kann er aber eine nützliche Anwendung finden.

> **Teil 5**
>
> **KURSÜBERSICHT**
>
> **Der CLI-Kurs erläutert Ihnen in mehreren Folgen den Umgang mit der Amiga-Benutzerschnittstelle CLI (Command-Line-Interface). Am Ende haben Sie alle wichtigen Befehle kennengelernt und können die Kommandos nach eigenen Wünschen einsetzen. Die folgende Kursübersicht zeigt Ihnen, in welchen Kursteilen welche CLI-Kommandos erklärt werden:**
>
> **TEIL 1** behandelte Befehle zum Anzeigen des Directories und Stellen der Uhr (DIR, LIST, CD, DATE, ENDCLI)
>
> **TEIL 2** erklärte Ihnen, wie Sie Dateien kopieren, löschen, umbenennen und Disketten formatieren können (FILENOTE, DELETE, RELABEL, RENAME, TYPE, COPY, DISKCOPY, INFO, FORMAT, INSTALL)
>
> **TEIL 3** zeigte unter anderem auf, wie Texte und Startdateien erstellt und bearbeitet werden (MAKEDIR, ED, PROMPT, WAIT, LOADWB, SETCLOCK, EXECUTE)
>
> **TEIL 4** befaßte sich mit Befehlen, die sich zum Einbau in eine Batch-Datei anbieten (ASSIGN, ECHO, FAILAT, IF, THEN, ELSE, ENDIF, STACK, PROTECT, BINDDRIVERS)
>
> **TEIL 5 setzt die Erklärung von Kommandos für eine Batch-Datei fort (RUN, EDIT, JOIN, LAB, QUIT, SKIP, ADDMEM, VERSION)**
>
> **TEIL 6** geht auf neue Spezialbefehle für Zusatzgeräte wie etwa Festplatten ein (SEARCH, NEWCLI, SORT, WHY, DJMOUNT, PREP, DPFORMAT, MOUNT, ADDBUFFERS, DISKCHANGE)
>
> **TEIL 7** schließt den CLI-Kurs mit den letzten Workbench-Befehlen ab (CHANGETASKPRI, DISKDOCTOR, FAULT, SETDATE, SETPREFS, STATUS)

Batchen mit Komfort

Mit Batch-Dateien läßt sich also eine Menge anfangen. Um aber ganz flexibel damit zu arbeiten, kann man beim Aufruf der Stapeldatei gleich einen Parameter übergeben, mit dem in der Befehlsdatei gearbeitet wird. Das könnte beispielsweise der Name eines Programms sein. Für das folgende Beispiel nehmen wir an, daß eine Befehlsfolge programmiert werden soll, die eine beliebige Datei mit einer bestimmten Folge von Befehlen »bearbeitet«. Wir möchten, daß im Beispiel die (beliebige) Datei zuerst einmal ins RAM kopiert wird, von dort auf Bildschirm und Drucker ausgegeben und auf eine in »df1« eingelegte Diskette kopiert wird. Dieser Vorgang ergibt zwar keinen rechten Sinn, doch dient er hervorragend zur Demonstration über den Aufbau einer flexiblen Stapeldatei.

Wir möchten erreichen, daß nur einmalig beim Aufruf der Batch-Datei der Name des Programms übergeben wird, mit dem wir den Vorgang durchführen. Da in der Befehlsdatei der Name des Programms sicher öfter verlangt wird, ist es das einfachste, einen Platzhalter dafür einzusetzen. Genau dies erreicht man mit der Definition ».KEY Name«.

Bei Angabe dieser Definition am Beginn der Datei wird dem Amiga mitgeteilt, daß hinter dem Namen der Batch-Datei noch ein weiterer Parameter folgt. Hinter dem Namen kann, durch einen Schrägstrich ge-

trennt, stehen, ob der Parameter nötig ist (/A). Ein Beispiel:

```
.key name/a
...
LOADWB
```

Nehmen wir an, der Name der Batch-Datei wäre »Test« und der des zu bearbeitenden Programms »xyz«. Über die Tastatur müssen Sie zum Aufruf dieser Stapeldatei

```
EXECUTE Test xyz
```

eingeben. Ab jetzt würde überall in der Batch-Datei, wo <Name> steht, der Platzhalter »xyz« eingesetzt. Für das obige Beispiel sieht die zu erstellende Stapeldatei so aus:

```
.key name/a
COPY <NAME> TO ram:
TYPE ram:<Name>
COPY ram:<Name> TO prt:
COPY ram:<Name> TO df1:
```

Anstelle von <Name> findet sich bei der Abarbeitung der mit dem Aufruf übergebene Name der zu bearbeitenden Datei. Es kann aber vorkommen, daß eine Befehlsfolge programmiert werden soll, bei der nicht unbedingt ein Name übergeben werden soll. Wir wollen annehmen, daß in diesem Fall ein vorher bestimmter Platzhalter eingesetzt wird. Die Definition ».DEF Name Name2« bestimmt, daß, wenn kein Name beim Aufruf übergeben wird, ein definierter Platzhalter Verwendung findet. In diesem Fall muß das »/A« hinter .KEY entfallen, damit nicht zwingend ein Übergabeparameter angefordert wird:

```
.key name
.def name xxx
COPY <NAME> TO ram:
TYPE ram:<Name>
COPY ram:<Name> TO prt:
COPY ram:<Name> TO df1:
```

Gibt der Anwender jetzt beim Aufruf mit EXECUTE keinen Namen mehr an, setzt der Amiga anstelle von <Name> die mit .DEF bestimmte Zeichenkette »xxx« ein.

Das RAM wird aktiv

Speziell für den Amiga 1000 sind einige RAM-Erweiterungen zum internen oder externen Anschluß erhältlich, die teilweise den Nachteil aufweisen, daß sie nicht autokonfigurierend sind. Das bedeutet, daß sie vom Betriebssystem beim Initialisieren nicht automatisch erkannt werden. Um diese Speichererweiterungen dem Amiga doch noch zugänglich zu machen, gibt es den sowohl im Direkt- wie auch im Programmodus einsetzbaren Befehl ADDMEM.

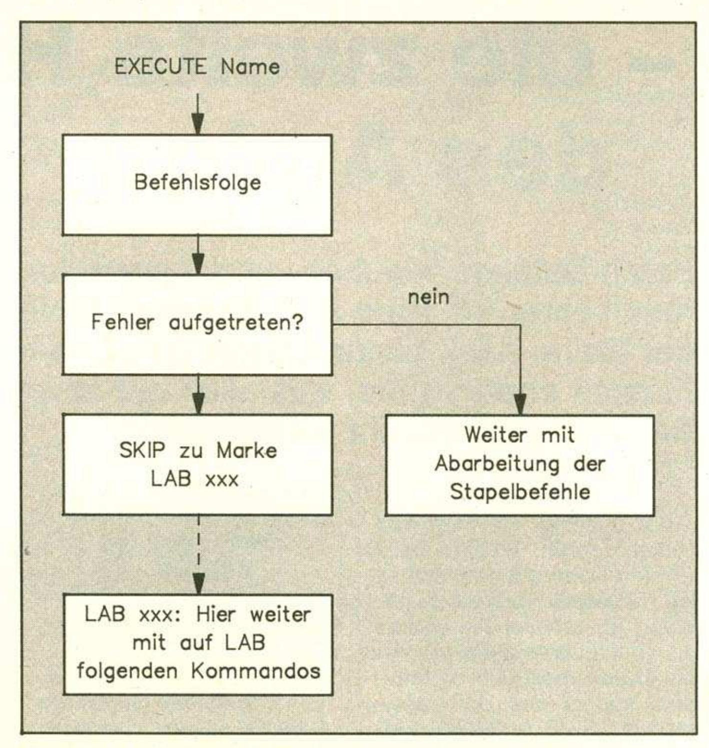


Bild. Die SKIP/LAB-Kombination funktioniert in etwa wie der von Basic her gewohnte Sprungbefehl GOTO

Er teilt dem Amiga mit, daß neuer Speicher einzubinden ist und an welcher Speicheradresse er beginnen soll. Dazu muß die Startadresse des neuen Speicherbereichs, also wo man ihn gerne haben will, und die Länge des RAM-Bereichs +1 angegeben werden. Zum Beispiel könnte der Aufruf für 1 MByte mehr Speicher, beginnend ab $200000, so lauten:

```
ADDMEM 200000 300000
```

Nach Bestätigung des Befehls mit <RETURN> steht dem Anwender der neue RAM-Bereich sofort zur Verfügung.

Von Zeit zu Zeit kann es nötig werden, daß mehrere Einzeldateien (etwa Textfiles) zu einem einzigen File verbunden werden müssen. Damit die Sache nicht zu umständlich wird, stellt das CLI den Befehl JOIN zur Verfügung. Er erlaubt, maximal 15 Einzeldateien zu einer neuen Gesamtdatei »zusammenzuschweißen«. Die einzige Bedingung beim Aufruf ist, daß die neue Datei einen anderen Namen als eine der Einzeldateien erhalten muß. Die zu verbindenen Dateien müssen in der Reihenfolge, wie sie verbunden werden sollen, dem Befehl folgen:

```
JOIN Text1 Text2 AS Text3
```

Die Dateien »Text1« und »Text2« werden zu der neuen Datei »Text3« vereint.

Einer der wohl am häufigsten gebrauchten Befehle im CLI ist RUN. Dieses äußerst nützliche Programm erlaubt es, vom CLI aus ein Programm (einen Task/Prozeß) zu starten, das abläuft, ohne das CLI zu blockieren. Um dies verständlicher zu machen, geben Sie bitte folgendes ein:

```
Preferences <RETURN>
```

Sobald der Preferences-Screen erschienen ist, legen Sie ihn mittels des Hintergrundschalters oben rechts nach hinten, so daß das CLI wieder sichtbar ist. Versuchen Sie, im CLI irgend einen Befehl einzugeben. Sie werden feststellen, daß das CLI keine Eingaben mehr verarbeitet. Holen Sie Preferences wieder in den Vordergrund und beenden das Programm mit dem Schließschalter links oben.

Nun probieren wir folgenden Befehl aus:

```
RUN Preferences
<RETURN>
```

Wiederholen Sie die Prozedur und geben einen Befehl im CLI ein. Sie sehen, das eben gestartete Preferences-Programm arbeitet im Hintergrund, während das CLI weiterhin Eingaben annimmt und verarbeitet. Der Befehl RUN startet also vom CLI aus ein Programm und gibt die Kontrolle anschließend zurück, so daß das Command Line Interface nicht blockiert ist. RUN ist ein äußerst nützlicher Befehl, der sowohl im Direkt- wie auch im Programmodus eingegeben werden kann und das Multitaskingsystem nicht an seiner Entfaltung hindert.

Aber RUN kann noch mehr: Mit einem einzigen Aufruf lassen sich auch mehrere Prozesse starten. Dazu muß nur zwischen jedem neuen Befehl ein »+« gesetzt werden.

```
RUN COPY df0:Disk.info
TO df0:Test.info+
DELETE df0:Test.info+
ECHO "Fertig!"
```

Während der Anwender nach dem Aufruf ungestört mit dem CLI weiterarbeiten kann, kopiert das Betriebssystem die Datei »Disk.info« in das File »Test.info« und löscht die Kopie anschließend. Erst durch die Meldung »Fertig!« bekommt der Benutzer mit, daß der Vorgang abgeschlossen ist.

Ein selten benutzter Befehl des CLI ist EDIT. EDIT ist ein wenig komfortabler Zeileneditor, der nur gestattet, vorhandene, sequenteill aufgebaute Dateien zu bearbeiten. Es ist nicht möglich, von EDIT aus neue Dateien zu erzeugen. Da er außerdem nur mit Binärdateien etwas anfangen kann und auch sonst relativ selten eingesetzt wird, gehen wir in diesem Kurs nicht näher darauf ein. Er wird nur der Vollständigkeit halber erwähnt. Hauptsächlich gedacht ist dieses zeilenorientiert arbeitende Programm wohl für Programmierer, die beispielsweise Maschinenprogramme oder ähnliches editieren will. Wer genaueres über diesen Befehl erfahren möchte, sei auf das Amiga-DOS-Handbuch verwiesen, das dem Amiga beiliegt.

Der letzte CLI-Befehl, der in diesem Kursteil besprochen wird, ist VERSION. Der Befehl ist ohne Parameter anzugeben. Er ziegt die Versionsnummern der Kickstart- und der Workbench an. So können Sie jederzeit erfahren, ob Sie die aktuellen Versionen besitzen (derzeit Kickstart 33.180/Workbench 33.60).

Sie sollten nun über genügend Wissen verfügen, um eigene Stapeldateien zu programmieren. Bis wir im nächsten Teil des Kurses auf neue Befehle für Zusatzgeräte (etwa Festplatten) eingehen, können Sie zur Übung mit den einzelnen Befehlen experimentieren und eigene Batch-Dateien erzeugen. (dm)

Unterstützende Literatur: Das Amiga-Handbuch, Markus Breuer, Markt & Technik-Verlag, 49 Mark

Basic für Aufsteiger

Während ein Programmierer sich nur ungern bei der Arbeit stören läßt, können solche »Unterbrechungen« beim Amiga den Bedienungskomfort erheblich erhöhen. Nutzen Sie diesen Sachverhalt in Ihren Basic-Programmen effektiv aus.

Anwender von Amiga-Basic-Programmen oder Programmierer dieser Sprache kennen die kleine OK-Box oberhalb des Bildschirms. Ein Fehler ist aufgetreten und damit ist der Programmablauf beendet. Eventuell vorher eingegebene Daten sind verloren. Ärgerlich? In den meisten Fällen ja. Amiga-Basic kennt daher einen Mechanismus, mit dem nach Auftreten eines Fehlers die Programmausführung in einen bestimmten Teil des Programms verzweigt. Hier könnten dann zum Beispiel vor dem Programmende wichtige Daten auf die Diskette gespeichert werden. Vielleicht läßt sich aber auch die Ursache des Fehlers durch weitere Eingaben beheben. Das Programm kann dann ohne einen Abbruch am Ort der Unterbrechung seine Tätigkeit wieder aufnehmen.

Dem Fehler auf der Spur

Damit ist das Stichwort für diesen Kursteil gefallen: Unterbrechungen. Eine Unterbrechung funktioniert ähnlich wie ein Unterprogrammsprung mit GOSUB. Im Unterschied zu dieser Technik wird die Verzweigung jedoch nicht durch einen Programmbefehl, sondern durch ein bestimmtes Ereignis hervorgerufen. Solche Ereignisse können das Betätigen der Maus, ein Menüklick oder aber das Auftreten eines Fehlers sein. In dem jeweiligen Unterprogramm wird dann auf das entsprechende Ereignis reagiert. Auf diese Weise läßt sich die von der Workbench gewohnte Benutzeroberfläche einfach in Basic-Programme integrieren. Auch das Abfangen von Softwarefehlern und eine Reaktion darauf durch Befehle des Programmierers sind mit diesen Funktionen einfach zu realisieren.

Fehlerlos ist sicherlich kein Programm. Als Programmierer müssen Sie alle Eventualitäten von Anfang an mit in Ihr Programm einplanen. Dies gilt besonders für die Fehler, die bei der Bedienung des Programms gemacht werden können. Um den Programmentwicklern die Arbeit zu erleichtern, bietet Amiga-Basic einerseits die Möglichkeit, Fehlermeldungen des Interpreters abzufangen, andererseits lassen sich auch neue Fehler definieren. Mit Hilfe des Befehls

```
ON ERROR GOTO Marke
```

können Sie dann auf das Auftreten eines dieser Fehler entsprechend reagieren. Doch woher wissen Sie, welcher Fehler aufgetreten ist? Dazu dient die vordefinierte Variable ERR. Jedem Fehler wird von Basic eine bestimmte Nummer zugeordnet. Diese Zuordnung können Sie dem Anhang B Ihres Basic-Handbuchs entnehmen. Wenn Sie dort jedoch nachschlagen, werden Sie feststellen, daß von den möglichen 255 Fehlercodes nur ein kleiner Teil eine Bedeutung hat, während den restlichen Fehlernummern der Fehlertext »Unprintable Error« zugeordnet ist. An dieser Stelle können Sie eingreifen: der Befehl

```
ERROR Nummer
```

gestattet es Ihnen, jeden Fehler zu simulieren, das heißt, falls Sie zum Beispiel in Ihrem Programm irgendwo die Zeile ERROR 2 plaziert haben, dann wird Ihr Programm genau dort mit der Fehlermeldung »Syntax Error« abgebrochen. Haben Sie nun aber zuvor irgendwo in Ihrem Programm die Befehlszeile ON ERROR GOTO Marke stehen, so verzweigt Ihr Programm erst zu der angegebenen Routine. Innerhalb dieser Routine ist es zweckmäßig, die Variable ERR auf ihren Wert zu überprüfen (in unserem Beispiel hat sie den Wert 2) und abhängig von diesem Wert können Sie nun versuchen, den Fehler zu beheben, oder aber zumindest eine Fehlermeldung ausgeben. Der Programmteil nach der durch den ON ERROR GOTO-Befehl angesprungenen Marke wird so lange abgearbeitet, bis Amiga-Basic auf den Befehl

```
RESUME
```

trifft. Den RESUME-Befehl gibt es in zwei Variationen. Sie können den RESUME-Befehl alleine angeben — dann kehrt Basic genau zu dem Befehl zurück, der den Fehler ausgelöst hat, und führt ihn nochmals durch. Sie können aber auch eine weitere Sprungmarke hinter RESUME angeben. In diesem Fall springt Basic zu dieser Marke. Dieses Verfahren ist jedoch mit größter Vorsicht zu genießen. Es gibt eigentlich nur zwei Fälle, in denen es angewendet werden sollte. Der erste ist ein Sprung an das Programmende und der zweite

Teil 4

KURSÜBERSICHT

Dieser Kurs hat das Ziel, Ihnen alles über die Programmierung des Amiga-Basic beizubringen. In den einzelnen Teilen des Kurses finden Sie folgende Themen behandelt:

das Schlüsselwort NEXT anstelle der Sprungmarke. Der Zusatz NEXT bewirkt, daß die Programmausführung direkt in der Zeile nach dem fehlerverursachenden Befehl fortfährt.

Als Beispiel für die Fehlerverarbeitung soll Ihnen Listing 1 dienen. Mit dem Programm lassen sich Texte in eine sequentielle Datei speichern. Für den Fall, daß die angegebene Datei bereits auf der Diskette existiert, sollen die in ihr enthaltenen Texte auf dem Bildschirm ausgegeben und weitere Texte daran angehängt wer-

Der Fehler Nummer 53

den. Zunächst wird die Fehlerunterbrechung aktiviert. Um nach der Eingabe des Dateinamens zu verhindern, daß die Datei mit einem Leerstring eröffnet wird, haben wir einen neuen Fehler erzeugt. Dieser hat die Nummer 42 und tritt immer dann auf, wenn bei der Eingabe lediglich die Taste <Return> betätigt wird. Natürlich kann dieser Fall auch einfacher abgefangen werden. Unser Beispiel soll Ihnen lediglich den Nutzen dieser Funktion erläutern.

```
ON ERROR GOTO Fehler
  CLS
Anfang:
  dn$="":INPUT "Dateiname:";dn$
  PRINT
  IF dn$="" THEN ERROR 42
  OPEN dn$ FOR INPUT AS #1
  IF Num=53 THEN OpenDatei
  IF Num=42 THEN Anfang
  WHILE NOT EOF(1)
    INPUT#1,e$
    PRINT e$
  WEND
  CLOSE 1

OpenDatei:
  PRINT
  PRINT "<*> beendet Eingabe"
  PRINT
  OPEN dn$ FOR APPEND AS #1
  INPUT "Satz: ";s$
  WHILE s$<>"*"
    PRINT#1,s$
    INPUT "Satz: ";s$
  WEND
  CLOSE 1
  END

Fehler:
  Num=ERR
  IF Num=53 THEN
    RESUME NEXT
  ELSEIF Num=42 THEN
    PRINT "Keine leere Eingabe"
    RESUME NEXT
  ELSE
    PRINT "Programmfehler!"
    PRINT "Programm beendet"
    END
  END IF
```

Listing 1. Fehler abfangen mit ON ERROR GOTO

Tritt beim Öffnen der Datei zum Lesen der Fehler Nummer 53 auf, dann ist die angegebene Datei nicht vorhanden. Das Programm kann in diesem Fall mit einer erneuten Eröffnung zum Schreiben und der Eingabe der Texte fortfahren. Tritt dieser Fehler allerdings nicht auf, werden alle Sätze der Datei gelesen und auf dem Bildschirm ausgegeben. Der in dieser Ausgabe veröffentlichte Checksummer benutzt übrigens eine solche Routine.

Zur Fehlerbearbeitung bleibt jetzt nur noch eine — allerdings wenig nützliche — Systemvariable zu besprechen. Die Variable ERL enthält beim Auftreten eines Fehlers die letzte Zeilennummer, die vor dem Auftreten des Fehlers abgearbeitet wurde. Leider muß diese Zeilennummer auch wirklich eine Nummer sein. Label werden nicht erfaßt. Da Sie aber wegen der mächtigen strukturierten Befehle des Amiga-Basic normalerweise auf Zeilennummern verzichten können, lassen sich nur wenige Einsatzmöglichkeiten für diese Variable finden.

Hinter der besprochenen Methode der Fehlerbearbeitung steckt ein allgemeines Programmierprinzip, das wir mit dem deutschen Wort Unterbrechungsverarbeitung vorgestellt haben. Wie üblich ist der entsprechende amerikanische Fachausdruck »event trapping« wesentlich anschaulicher. Denn dies ist genau die Absicht. Ein Programm zu schreiben, welches darauf wartet, daß ihm ein bestimmtes Ereignis in die Falle geht. Sobald dieses Ereignis eintritt, läßt Ihr Amiga alles stehen und liegen, um es sofort zu bearbeiten. Ist er damit fertig, kehrt er wieder zum eigentlichen Programm zurück, um es an der Stelle fortzusetzen, an der es unterbrochen wurde.

Es gibt viele Ereignisse, die eine derartige Bearbeitung erfordern. Zwei davon sind Ihnen schon bekannt: Zum einen die oben besprochene Fehlerbehandlung, zum anderen die bei den Animationsbefehlen besprochene Verarbeitung von Zusammenstößen grafischer Objekte. Weitere mögliche Ereignisse sind das Verstreichen einer gewissen Zeitspanne oder der Druck auf eine Maustaste. Besonders für die Mausbedienung bietet Amiga-Basic mächtige Funktionen, die eine einfache Maus- und Menüsteuerung erlauben.

Wie Sie ja schon von der Workbench gewohnt sind, besitzen die beiden Maustasten unterschiedliche Funktionen. Die rechte Maustaste dient zur Auswahl von Pull-Down-Menüs, während mit der linken Maustaste durch Anklicken Programme geladen beziehungsweise Icons oder Gadgets selektiert werden. Diese unterschiedliche Bedeutung der beiden Maustasten kommt auch in Amiga-Basic zum Tragen. Die beiden Befehle

```
ON MOUSE GOSUB Marke
ON MENU GOSUB Marke
```

machen dem Interpreter die Zeilennummer bekannt, zu der er nach dem entsprechenden Ereignis verzweigen soll. Wir wollen uns zunächst mit dem MOUSE-Befehl beschäftigen, der sich auf die linke Maustaste bezieht. Falls Sie in Ihrem Programm am Anfang den Befehl »ON MOUSE GOSUB Maus« gefolgt von

```
MOUSE ON
```

setzen, wird jedesmal, wenn Sie die linke Maustaste betätigen, zur Sprungmarke »Maus« verzweigt. Hier müssen nun diejenigen Befehle stehen, die die Verarbeitung dieses Ereignisses übernehmen sollen. Mit Hilfe der Funktion

```
MOUSE (n)
```

können Sie sich, je nach dem Wert von n, über den augenblicklichen Zustand der Maus informieren. Die wichtigste dieser Funktionen ist MOUSE(0). Diese bringt für die restlichen sechs Funktionen MOUSE(1) bis MOUSE(6) die Position des Mauszeigers auf den aktuellen Stand. Dabei erhalten die ungeradzahligen MOUSE-Funktionen die x-Koordinate und die geradzahligen MOUSE-Funktionen jeweils die y-Koordinate des Mauszeigers. Doch wodurch unterscheidet sich zum Beispiel die mit MOUSE(1) ermittelte x-Position von der mit MOUSE(3) ermittelten? Am besten führen wir dazu ein kleines Experiment durch. Tippen Sie Listing 2 ab und starten Sie das Programm. Bewegen Sie jetzt Ihren Mauszeiger über den Bildschirm, ohne dabei eine Maustaste zu betätigen. Wie Sie sehen, verändern sich nur die Werte in der ersten Spalte, während alle anderen Werte gleich bleiben. Damit haben wir auch schon unser erstes Ergebnis. Die Funktionen MOUSE(1) und MOUSE(2) zeigen die bei einem MOUSE (0)-Aufruf vorliegende x- beziehungsweise y-Position des Mauszeigers unabhängig davon, ob Sie eine Maustaste gedrückt haben oder nicht. Nun betätigen Sie die linke Maustaste und halten sie fest. Jetzt nehmen alle drei Spalten (beinahe) die gleichen Werte an. Also haben wir herausgefunden, daß die Funktionen MOUSE(3) bis MOUSE(6) auf das Betätigen der Auswahltaste reagieren. Ziehen Sie nun die Maus mit festgehaltener Auswahltaste über den Bildschirm.

```
f$ = "###"
LOCATE 1,7  : PRINT "MOUSE(1/2)"
LOCATE 1,27 : PRINT "MOUSE(3/4)"
LOCATE 1,47 : PRINT "MOUSE(5/6)"
LOCATE 2,1 : PRINT "x = "
LOCATE 3,1 : PRINT "y = "
WHILE 1
  dummy = MOUSE(0)
  LOCATE 2,10 : PRINT USING f$;MOUSE(1)
  LOCATE 3,10 : PRINT USING f$;MOUSE(2)
  LOCATE 2,30 : PRINT USING f$;MOUSE(3)
  LOCATE 3,30 : PRINT USING f$;MOUSE(4)
  LOCATE 2,50 : PRINT USING f$;MOUSE(5)
  LOCATE 3,50 : PRINT USING f$;MOUSE(6)
WEND
END
```

Listing 2. Ausgabe der Mausfunktionen

Die Maus in der Falle

Wie erwartet ändert sich die erste Spalte. Aber auch in der dritten Spalte werden neue Werte angezeigt, während die zweite Spalte unverändert bleibt. Wir müssen das so interpretieren: Die MOUSE-Funktionen 3 und 4 geben die Koordinaten an, bei denen die Auswahltaste gedrückt wurde, während die MOUSE-Funktionen 5 und 6 die Koordinaten bezeichnen, bei denen entweder die Auswahltaste losgelassen wurde, oder aber — falls die Auswahltaste noch gedrückt ist — an welcher Position sich die Maus augenblicklich befindet.

Damit sind wir so weit, daß Sie sich an Listing 3 wagen können. In der ersten Zeile befehlen wir Amiga-Basic, daß es

sich bei jeder Betätigung der Auswahltaste sofort an die mit dem Label »Maus« bezeichnete Stelle begeben soll. Die Aktivierung dieser Unterbrechungsverarbeitung haben wir durch den Befehl

```
MOUSE ON
```

programmiert. Ebenso wird mit dem Timer verfahren, wobei das Argument (hier eins) angibt, in welchem Sekundenintervall dieser Aufruf durchgeführt werden soll. Dann soll ein Kreis mit dem Radius r und dem Mittelpunkt (xpos,ypos) gezeichnet werden. Falls Sie sich über die etwas seltsame WHILE...WEND-Schleifenkonstruktion wundern: Hinter dem WHILE-Befehl steht normalerweise eine logische Bedingung wie etwa »x < 100« und die WHILE-Schleife wird so lange ausgeführt, bis diese Bedingung unwahr ist. Nun ist es aber eine Eigenschaft von Basic, allen von Null verschiedenen Zahlen den logischen Wert »wahr« zuzuordnen, während die Zahl Null logisch »falsch« entspricht. Damit ist der Wert eins gleichbedeutend mit »wahr« und so haben wir eine Endlosschleife konstruiert.

Wäre Ihr Amiga ein Heimcomputer ohne die Fähigkeiten des Multitasking, wäre der im Schleifenkörper enthaltene Befehl SLEEP sinnlos, da sich Basic im Moment sowieso in einer Endlosschleife befindet. Der SLEEP-Befehl führt jedoch dazu, daß Ihr Basic-Programm solange angehalten wird, bis ein Unterbrechungsereignis eintritt. Gegenüber der normalen Endlosschleife hat dies den Vorteil, daß Ihr Programm anderen, gleichzeitig laufenden Programmen mehr Laufzeit zur Verfügung stellt. Das Basicprogramm wartet in diesem Zustand also nur darauf, daß Sie die Maustaste bedienen. (In der heutigen Zeit bringen wohl nur noch Computer hierfür die Geduld auf.)

Welche Funktion hat das Programm nun? In der ersten Zeile nach dem Label »Maus:« wird der augenblickliche Zustand der Maustaste überprüft. Das Programm unterscheidet zwei Zustände. Erstens: die Maustaste wurde ein-, zwei- oder dreimal kurz betätigt. Zweitens: die Maustaste bleibt längere Zeit gedrückt. Da Ihr Amiga sehr eilfertig ist, werden Sie keine Zeit haben, den Finger von der Maus zu nehmen. Schon wird angezeigt, daß die Auswahltaste gegenwärtig gedrückt ist. Um Ihnen die Zeit zu geben, den Finger von der Maus zu nehmen, folgt nun eine Verzögerungsschleife. Danach wird die Maustaste mit MOUSE(0) nochmals abgefragt. An dieser Stelle sollten

Der Timer wird aktiviert

Sie nun wissen, was die Zahlenwerte, die Sie durch diese Funktion erhalten, bedeuten. Liefert die Funktion MOUSE(0) als Ergebnis eine Null, so heißt dies, daß die Maus seit dem letzten Aufruf von MOUSE(0) nicht betätigt wurde. Mit einer Eins teilt uns der Interpreter einen einmaligen Tastendruck mit. Die Auswahltaste war jedoch zur Zeit des Aufrufs nicht mehr gedrückt. Analog bedeutet eine zwei, daß die Maustaste zweimal betätigt wurde und im Moment nicht niedergehalten wird. Eine drei gibt an, daß die Taste mehr als zweimal betätigt, danach aber losgelassen wurde. Ebenso sind negative Zahlen eine Information über die Anzahl der Mausklicks. Allerdings mit dem Unterschied, daß die Taste auch im Moment des Aufrufs noch niedergehalten wurde.

```
ON MOUSE GOSUB Maus
ON TIMER(1) GOSUB uhr

xpos = 300  :REM Setzen des
ypos = 100  :REM   Kreises
r = 10
Farbe=1
CIRCLE (xpos,ypos),r,Farbe
MOUSE ON :REM Unterbrechungs-
         :REM verarbeitung
TIMER ON :REM   aktivieren

WHILE 1  :REM Amiga-Basic
  SLEEP  :REM   schläft jetzt
WEND

Maus:
  Flag%=0
  Status%=MOUSE(0)
  LOCATE 2,10
  PRINT "Status 1: ";Status%
  FOR i%=1 TO 250 : NEXT i%
  Status%=MOUSE(0)
  LOCATE 3,10
  PRINT "Status 2: ";Status%

  IF Status%<0 THEN
    WHILE Status%<0
      CIRCLE (xpos,ypos),r,0
      xpos=MOUSE(1)
      ypos=MOUSE(2)
      CIRCLE (xpos,ypos),r,Farbe
      Status%=MOUSE(0)
      LOCATE 3,10
      PRINT "Status 2: ";Status%
      Flag%=0
    WEND
  ELSE
    Farbe=Status%
  END IF
RETURN

uhr:
  LOCATE 2,60 : PRINT DATE$
  LOCATE 3,61 : PRINT TIME$
  Flag%=Flag%+1
  IF Flag%=10 THEN
    LOCATE 10,10
    PRINT "... Zeit abgelaufen"
    END
  END IF
RETURN
```

Listing 3. Ein Programm reagiert auf die Maus

Bewegen Sie die Maus nun an irgend eine Stelle des Bildschirms. Halten Sie die Maustaste niedergedrückt. Der Kreis verändert seine Position und befindet sich nun unterhalb des Mauszeigers. Durch Verschieben der Maus läßt sich der Kreis an eine beliebige Stelle des Bildschirms plazieren. Die Bewegung stoppt mit Loslassen der Auswahltaste. Durch eine kurze Betätigung derselben Taste können Sie die Farbe beim nächsten Zeichenvorgang bestimmen. Ein einmaliger Klick bedeutet, daß der Kreis in Weiß gezeichnet wird; zwei- und dreimalige Klicks verändern die Farbe in Schwarz beziehungsweise Rot (Standardeinstellung der Workbench vorausgesetzt).

Das Menü zum Dessert

Das Programm beendet selbsttätig, wenn Sie zehn Sekunden die Maus nicht betätigt haben. Hierfür befindet sich in der Unterbrechungsroutine für den Timer ein Zähler, der bei jedem Einsprung um eins erhöht wird. Hat er den Wert zehn erreicht, endet das Programm. Jede Mausbetätigung muß diesen Zähler natürlich wieder auf Null setzen. Damit ist das Programmende genau zehn Sekunden nach dem letzten Bedienereingriff gewährleistet.

Die rechte Taste der Maus, die sogenannte Menütaste, hat eine gänzlich andere Aufgabe. Wenn Sie die Menüleiste mit dem Mauszeiger anfahren und dann die Menütaste drücken, erscheint die Menüleiste. Wählen Sie nun eine dieser Überschriften mit einem Mausklick aus, so klappt eine Liste von Auswahlpunkten in den Bildschirm. Genau dieses Verhalten können Sie auch von Basic aus steuern. Als Beispiel steht Ihnen Listing 4 zur Verfügung.

Zunächst muß der Interpreter wissen, wieviele Menüs Sie anlegen wollen. Die Anzahl der Menüs ist allerdings auf zehn beschränkt. Mehr Überschriften passen in der Regel auch nicht in die Menüleiste. Eine weitere Beschränkung gilt für die Anzahl der Unterpunkte.

Ein Menü darf maximal 19 davon haben. Die Unterpunkte dürfen selbst keine weiteren Untermenüs enthalten. Das Programm aus Listing 4 beschränkt sich auf zwei Hauptmenüs. Mangels einer besseren Idee haben wir sie einfach »Menü 1« und »Menü 2« genannt. Eingerichtet werden diese beiden Menüs jeweils durch den MENU-Befehl. Mit der ersten Nummer im Anschluß an das Schlüsselwort MENU machen Sie dem Basic-Interpreter bekannt, an welcher Stelle der Menüleiste das Menü stehen soll. Mit einer Eins an dieser Stelle legt der Interpreter dieses an die linke Seite und mit einer zehn wird es an der rechten Seite plaziert. Dazwischenliegende Werte legen die Position entsprechend zwischen die beiden Enden der Menüleiste fest.

Das Menü in Geisterschrift

Der zweiten Zahl nach dem MENU-Befehl kann Ihr Amiga entnehmen, an welcher Stelle innerhalb des Hauptmenüs der angegebene Unterpunkt plaziert werden soll. Eine Null an dieser Stelle besagt, daß es sich bei der angegebenen Zeichenkette um den Menütitel selbst handelt. Mit Werten von 1 bis 19 kann nun jeder Menüunterpunkt im Menü positioniert werden.

Der Sinn des dritten Wertes wird von vielen Programmierern noch unterschätzt. Mit ihm läßt sich der entsprechende Menüpunkt aktivieren oder deaktivieren. Setzen Sie hier eine Eins ein, erscheint in Ihrem Menü der entsprechende Schriftzug in normaler Schrift. Das Menü ist aktiviert und kann durch einen Mausklick angewählt werden. Es gibt allerdings Situationen während des Programmablaufs, bei denen der Programmierer den Bediener an der Auswahl eines Unterpunktes oder ganzen Menüs hindern möchte. Dies kann er durch eine Null als dritten Wert erreichen. Das Menü erscheint dann in der Ihnen sicher bekannten Geisterschrift. Ein derart gekennzeichneter Menüpunkt wird zwar noch angezeigt, die Unterbrechungssteuerung reagiert allerdings nicht mehr auf eine Auswahl. Wird das Hauptmenü selbst so gekennzeichnet, dann ist das gesamte Menü nicht mehr für eine Auswahl verfügbar.

Eine weitere Hilfe für den Bediener ist die Wahl des Wertes zwei an dieser Stelle. Der Amiga setzt dann vor den Menütext einen Haken. Dabei sollten Sie allerdings beachten, daß dieser Haken Platz beansprucht. Zwei Leerzeichen vor dem eigentlichen Text genügen für diesen Zweck. Sinn ergibt diese Hakentechnik bei Menüpunkten, die eine Schalterfunktion haben. In einer Textverarbeitung lassen sich so beispielsweise aktivierte Schriftattribute (fett, kursiv, unterstrichen...) kennzeichnen.

Das Programm (Listing 4) beginnt mit zwei Datazeilen. Diese enthalten die einzelnen Menütexte. Am Anfang der Zeile befindet sich der Menütitel; daran anschließend die Unterpunkte. Jedes Menü wird mit einem »*« abgeschlossen. Das Ende der Daten zeigt »#« an. Die folgende While-Schleife aktiviert die Menüs. Durch eine Änderung der Datazeilen (Hinzufügen/Löschen von Texten) lassen sich ohne Veränderung der Programmanweisungen die Menüs einfach umstellen oder ergänzen.

Nach einer Aktivierung der Unterbrechungsverarbeitung schicken wir den Amiga wie gehabt in den Schlaf. Mit den Anweisungen

```
ON BREAK GOSUB Ende
BREAK ON
```

befehlen wir dem Interpreter, auf die Marke »Ende:« zu verzweigen, wenn der Bediener das Programm mit <Amiga.> oder <CTRL C> beendet. Damit ist gewährleistet, daß auch in diesen Fällen mit

```
MENU RESET
```

die Menüs wieder aus der Menüleiste entfernt werden. Der Befehl ON BREAK.. muß jedoch mit Vorsicht gehandhabt werden. Sie können damit leicht eine Endlosschleife konstruieren, aus der Sie nur durch die Tastenkombination <CTRL Amiga Amiga> wieder herauskommen.

Nach Auswahl eines Menüpunktes verzweigt der Interpreter in die mit »Wahl:« gekennzeichnete Routine. Dort ermittelt das Programm mit der Funktion MENU(0) die Nummer des angewählten Menüs und durch MENU(1) den entsprechenden Unterpunkt. Die Verteilung auf die zu jedem Menüpunkt gehörende Unterbrechungsroutine ist mit zwei ON GOSUB-Anweisungen realisiert. Aus Platzgründen münden alle Verzweigungen allerdings wieder in einem Programmteil zusammen. Mit Hilfe des Feldes »Flag%« wird dort lediglich ein Schalter umgelegt und das Menü entsprechend mit Haken versehen oder aber der Haken gelöscht.

```
DATA "Menu 1"," Punkt 1.1"," Punkt 1.2"," Punkt 1.3","
Ende",*
DATA "Menu 2"," Punkt 2.1"," Punkt 2.2"," Punkt 2.3",*
DATA #

i%=1
READ mt$
WHILE mt$<>"#"
  MENU i%,0,1,mt$
  j%=1
  READ mup$
  WHILE mup$<>"*"
    MENU i%,j%,1,mup$
    Flag(i%,j%)=1
    j%=j%+1
    READ mup$
  WEND
  i%=i%+1
  READ mt$
WEND

ON MENU GOSUB Wahl
MENU ON
ON BREAK GOSUB Ende:
BREAK ON

WHILE 1  :REM Basic schläft nun
  SLEEP  :REM und wartet auf
WEND     :REM Mausklick

Ende:
MENU RESET
END

Wahl:
  m% = MENU(0) : p% = MENU(1)
  ON m% GOSUB Menu1,Menu2
  RETURN

Menu1:
  ON p% GOSUB Menu11,Menu12,Menu13,Ende
  RETURN

Menu2:
  ON p% GOSUB Menu21,Menu22,Menu23
  RETURN

Menu11:
Menu12:
Menu13:
Menu21:
Menu22:
Menu23:
  IF Flag(m%,p%)=1 THEN
    Flag(m%,p%)=2
  ELSE
    Flag(m%,p%)=1
  END IF
  MENU m%,p%,Flag(m%,p%)
  RETURN
```

Listing 4. Unterbrechungsverarbeitung mit Menüs

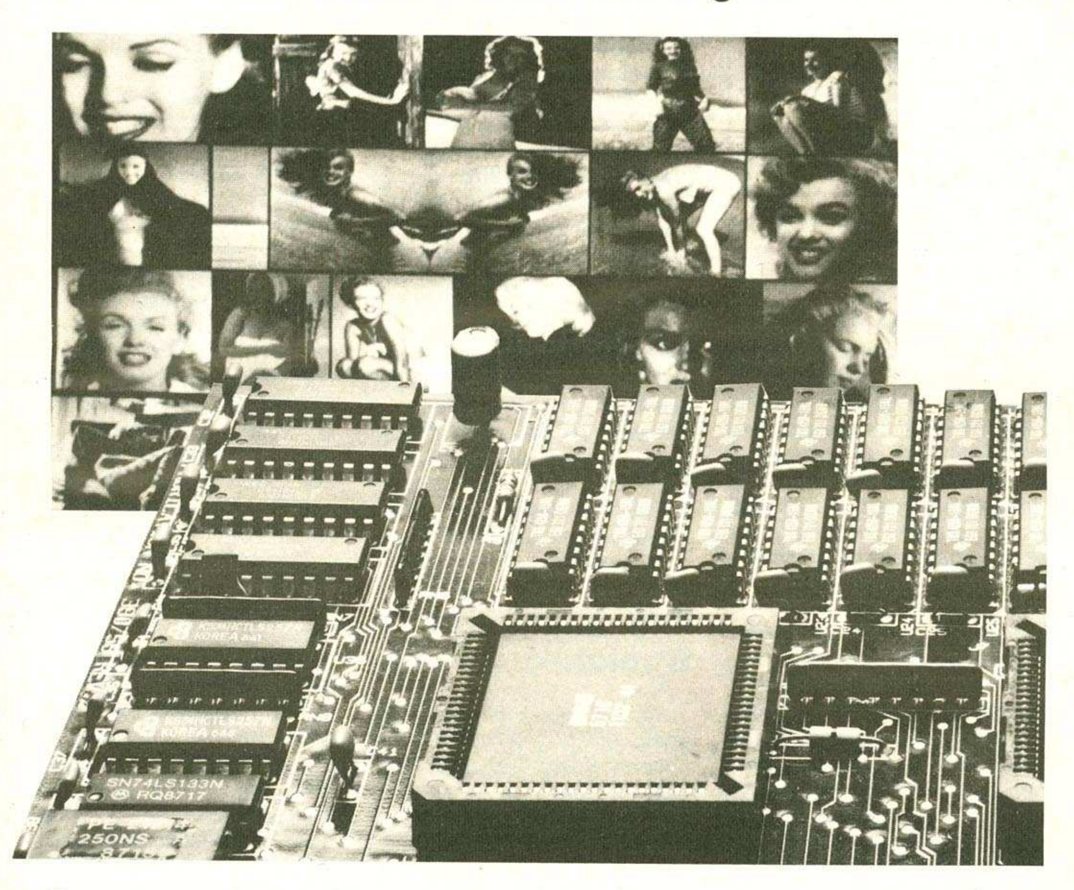
Zeit sparen mit Standardmenüs

Wenn Sie öfter Programme mit intensiver Nutzung der Menüs schreiben, oder aber ein Programm entwickeln, welches im Laufe der Zeit um weitere Menüfunktionen erweitert werden soll, lohnt sich das Anlegen eines Standardprogramm-Moduls. Dieses Modul enthält DATA-Zeilen für alle möglichen Menüpunkte (Leerstrings), die Sprungverteiler sowie Ansprungstellen für diese Menüpunkte mit anschließendem RETURN. Benötigen Sie nun ein Menü, kann das Modul in das aktuelle Programm eingefügt werden. Nun brauchen Sie nur noch die DATA-Zeilen und Unterprogramme ergänzen und fertig ist die Menüverwaltung.

Damit sind wir am Ende dieses Kursteiles. Die besprochene Thematik bietet sehr viel Raum für eigene Experimente. Sie werden beim Ausprobieren dieser Technik begeistert feststellen, wie einfach sich Programme damit strukturieren lassen. (Jürgen Singer/pa)

Grafik auf dem Amiga

Die fantastischen Grafikeigenschaften des Amiga sind einer leistungsfähigen Hardware zu verdanken. Dieser Kurs vermittelt Ihnen das Wissen, um mit der Steuerung dieser Hardware verblüffende Effekte am Bildschirm zu erzeugen.



Computergrafiken sind nicht nur eine reine Mode-Erscheinung, sondern haben in den letzten Jahren ganze Berufsbilder verändert oder neu geschaffen. In der breiten Öffentlichkeit begann es mit so spektakulären Anwendungen wie Computerspielen oder der Fernsehanimation. Heute wird die Computergrafik professionell für Konstruktionszeichnungen, Desktop Publishing und in vielen Bereichen der Werbung verwendet. Grafikcomputer wie der Amiga sind durch deren günstigen Preis einem großen Anwenderkreis zugänglich. Der Hobbyanwender kann sich mit einem vor ein paar Jahren noch undenkbar geringen finanziellen Aufwand mit diesem neuen Medium vertraut machen.

Wir werden uns in diesem Kurs hauptsächlich mit denjenigen Elementen im Amiga beschäftigen, welche die Grafikfähigkeiten dieses Computers erst möglich machen, und das ist in erster Linie die Hardware des Amiga. Beginnen wir mit dem wohl wichtigsten Baustein (Chip) im Computer: der CPU (Central Processing Unit oder Zentraleinheit). Die CPU ist derjenige Chip, der die eigentlichen Anwenderprogramme ausführt, also rechnet, Entscheidungen trifft oder Speicherplätze mit unterschiedlichen Werten belegt. Im Amiga ist das der Mikroprozessor MC 68000 von Motorola. Um die Leistungsfähigkeit des Computersystems zu erhöhen, haben die Entwickler mehrere Spezialchips zur Unterstützung der CPU eingebaut. Diese Custom-Chips geben dem Amiga seine Fähigkeiten nicht nur, was die grafischen und akustischen Anwendungen angeht, sondern ermöglichen beispielsweise erst die Multitaskingfähigkeiten des AmigaDOS. Der Vorsatz »Custom« bedeutet hier, daß diese Chips im Gegensatz zu Standardchips speziell für einen Computer entwickelt worden sind und auch nur für diesen hergestellt werden.

Agnus, Paula und Denise — das sind die drei Spezialchips — sind sogenannte »Coprozessoren«, die ihrer Arbeit unabhängig vom Hauptprozessor nachgehen. Was bedeutet nun »unabhängig« in diesem Zusammenhang? Ein Beispiel soll das verdeutlichen: Sowohl der Amiga als auch der legendäre C 64 von Commodore haben einen Soundchip. Beim Amiga ist das Paula und beim C 64 ist das ein SID genannter Baustein. Um auf dem C 64 Töne auszugeben, mußte der Prozessor die dazu notwendigen Daten aus seinem Speicher holen und Wert für Wert dem SID übergeben. Das kostete wertvolle Prozessorzeit. Beim Amiga teilt der Prozessor dem Soundchip lediglich mit, wo sich die Daten im Speicher befinden, und Paula holt sie sich dann selbst ab.

Die CPU bereitet also eine Aufgabe für die Custom-Chips lediglich vor. Sie versorgt die Coprozessoren mit den notwendigen Informationen, startet die Ausführung und kümmert sich dann um andere Dinge. Währenddessen wird der Coprozessor tätig und erledigt die ihm aufgetragene Arbeit. Ist er fertig damit, unterbricht er die Befehlsausführung des Prozessors in der Regel durch ein entsprechendes Signal und kann dann wieder neu mit Arbeit versorgt werden. Solche Unterbrechungen (Interrupts) spielen in der Computertechnik eine große Rolle.

Meister und Sklave

Dieses (fast) parallele Arbeiten wird also möglich, weil die Custom-Chips wie auch der Prozessor »direkt« auf den Speicher zugreifen können. Man bezeichnet diese Arbeitsweise der Hilfschips auch als DMA (direkt memory access -> direkter Speicherzugriff). Dieser direkte Speicherzugriff ist allerdings auf die unteren 512 KByte des Speichers beschränkt. Auf zusätzliche Speichererweiterungen kann nur die CPU selbst zugreifen.

Doch nun zu den einzelnen Chips. Da Paula hauptsächlich für die Tonerzeugung von Bedeutung ist, wollen wir uns mit diesem Chip nicht näher auseinandersetzen. Agnus enthält unter anderem einen sogenannten Adreßgenerator, der den DMA-Zugriff aller Coprozessoren auf den Speicher ver-

Teil 1

KURSÜBERSICHT

In diesem Kurs stellen wir Ihnen die Arbeitsweise der für die Bilddarstellung wichtigen Chips des Amiga vor. In den einzelnen Folgen des Kurses finden Sie folgende Themen behandelt:

TEIL 1 (Einführung): Die Spezialchips im Amiga; Bilddarstellung durch Punkte; Arbeitsweise von Monitor und Fernseher; Farbbildung durch Bitplanes

TEIL 2 (Playfields): Die Playfield-Hardware; der Datenzugriff durch Denise; Dual-Playfields; Scrolling; Hold & Modify; Interlace

TEIL 3 (Sprites): Datenstruktur, Positionierung und Bewegung von Sprites; virtuelle Sprites; Sprite-Attach

TEIL 4 (Blitter/Copper): Arbeitsmodi des Blitter; Programmierung des Copper; die Kontrollhardware

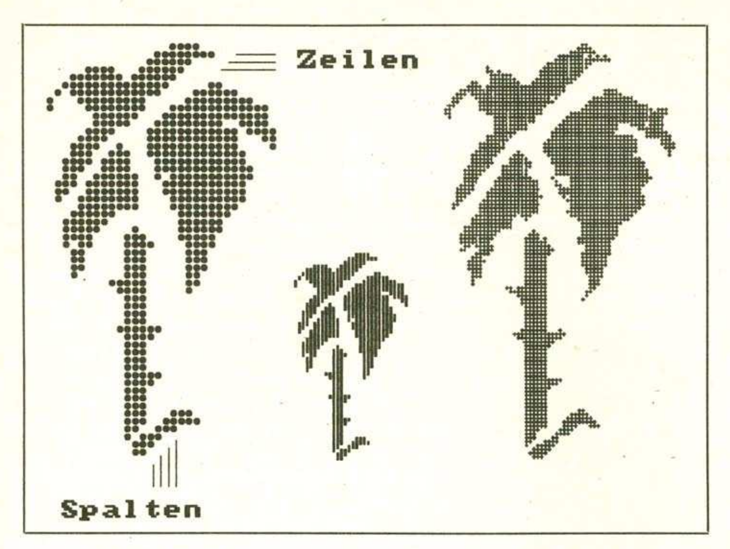


Bild 1. Die Darstellung eines Bildes durch viele Punkte

waltet. Zwei Coprozessoren innerhalb des Agnus-Chips, der Blitter und der Copper, tragen besonders zu schnellen bewegten Grafiken auf dem Bildschirm bei. Der Blitter ermöglicht ein schnelles Verschieben und Kombinieren von Speicherblöcken. Diese Technik wird beim Bewegen größerer Objekte (Bobs) auf dem Bildschirm angewendet. Der Copper hingegen arbeitet eng mit dem Elektronenstrahl des Monitors zusammen. Er erkennt, an welcher Zeile oder Spalte der Strahl sich gerade befindet und kann an bestimmten Positionen durch Beeinflussung anderer Chips besondere Aktionen auslösen. Denise ist der für die Grafikprogrammierung wohl wichtigste Chip des Amiga. Bevor wir uns mit seinen Funktionen näher beschäftigen, unternehmen wir noch einen kleinen Ausflug in die Praxis der Bilddarstellung:

Wenn Sie einmal ein Bild nicht mit einem Computer, sondern mit so »altertümlichen« Mitteln wie Pinsel oder Bleistift erstellen, so besteht Ihre Grafik in der Regel aus Strichen. Für die technische Verarbeitung oder Übertragung solcher Bilder ist ein anderes Verfahren entwickelt worden: die Darstellung durch Punkte (Bild 1 links). Wenn Sie Bild 1 mit dem beim Lesen üblichen Abstand betrachten, können Sie die einzelnen Punkte erkennen. Je weiter Sie sich jedoch von dieser Darstellung entfernen, desto weniger ist das Auge in der Lage, die einzelnen Punkte zu unterscheiden: aus nebeneinanderliegenden Punkten werden Striche — der gewohnte Eindruck eines Bildes entsteht. Durch Verminderung des Punktzwischenraums kann der Eindruck eines »sauberen« Bildes auch schon bei relativ nahem Betrachtungsstandpunkt entstehen. Dadurch wird das Bild aber kleiner (Bild 1, Mitte). Ist dieser Effekt nicht erwünscht, so müssen mehr Punkte pro Bild eingesetzt werden (Bild 1 rechts). Man bezeichnet diese Punktdichte auch als Auflösung. Je höher die Auflösung eines Bildes ist, desto mehr Punkte werden für seine Darstellung eingesetzt.

```
001111110000000
000111100000000
000111000000000
000111000000000
000111000000000
000111000000000
000111110000000
000111100000000
000111000000000
000111000000000
000111000000000
000111000111000
000111001111100
000110011100000
000101110000000
000011100000000
000011000000000
```

Bild 2. Bitplane-Ausschnitt der Palme aus Bild 1

Der Amiga kann Bilder in verschiedenen Auflösungen darstellen. In der »normalen« Auflösung sind 320 x 200 Punkte darstellbar. Was bedeutet nun »320 x 200 Punkte«? Schauen wir uns dazu noch einmal Bild 1 an. Durch die Zerlegung der Grafik in Punkte lassen sich eindeutig untereinander und nebeneinander angeordnete Punktreihen unterscheiden. Die untereinander, also waagerecht angeordneten Punktreihen nennt man Zeilen. Die nebeneinander (vertikal) angeordneten Punktreihen heißen Spalten. Um eine solche Auflösung zu beschreiben, benutzt man den Ausdruck 320 Zeilen mal (zu je) 200 Spalten — oder einfach: 320 x 200 Punkte (oder Pixel).

Woher weiß der Amiga aber, welche Punkte er auf dem Bildschirm setzen soll und welche nicht? Unter einem gesetzten Punkt verstehen wir einen Punkt in der Vordergrundfarbe. Auf dem Papier ist die Vordergrundfarbe Schwarz und die Hintergrundfarbe in der Regel Weiß. Auf dem Bildschirm ist es meist umgekehrt: die dunklere Farbe ist der Hintergrund. Da Sie diese Einstellung verändern können, wollen wir für unsere Betrachtungen von der Standardeinstellung der Workbench ausgehen: Vordergrund weiß; Hintergrund blau. Ein gesetzter Punkt ist also weiß.

Kommen wir auf die eben formulierte Frage zurück. Im Speicher des Amiga reserviert das Betriebssystem einen Bereich für die Bilddarstellung. Man kann sich diesen Bereich als einen »Haufen« winziger Speicherzellen vorstellen, die nur zwei verschiedene Zahlen speichern können: Null oder Eins. Dieser Speicherbereich ist genau wie unser Bild zeilenbeziehungsweise spaltenweise organisiert.

Das Abbild im Speicher

Jede Eins in diesem Bildspeicher entspricht einem gesetzten Punkt auf dem Bildschirm. Eine Null — die logische Konsequenz — bedeutet, daß der entsprechende Punkt auf dem Bildschirm in der Hintergrundfarbe abgebildet ist. Der Bildschirmspeicher (auch Frame Buffer oder Bitmap genannt) ist also im Prinzip ein genaues Abbild der Bildschirmdarstellung, bestehend aus binären Einsen und Nullen. Bild 2 zeigt einen Ausschnitt der Bitmap des Motivs aus Bild 1 in Form von Nullen und Einsen.

Die in einer solch winzigen Speicherstelle gespeicherte Informationseinheit (eben Null oder Eins) nennt man übrigens Bit. Dieses Wort leitet sich ab von der Bezeichnung »binary digit« (binäre Ziffer). Das Bit spielt in der Computertechnik eine große Rolle. Wer sich mit Computer im allgemeinen oder mit der Grafikdarstellung im besonderen beschäftigt, sollte sich Informationen zu diesem Thema beschaffen. Wir wollen aus Platzgründen darauf nur dann eingehen, wenn es das Verständnis der in diesem Kursus vorgestellten Sachverhalte erleichtert.

Der Computer muß sich also die Bildinformationen aus seinem Speicher holen. Diese Arbeit erledigt Denise. Auf diesen Chip werden wir noch genauer eingehen. Zunächst ist er erst mal ein schwarzer Kasten (Black Box), der sich einerseits die Nullen und Einsen aus dem Speicher holt und andererseits die Impulse für die Steuerung des Bildaufbaus an den Monitor weiterleitet.

Denise beginnt bei der Übersetzung der Bits in der linken oberen Ecke des Frame Buffers und leitet die Informationen für die Bilddarstellung Zeile für Zeile an die entsprechenden Schaltungen des Monitors weiter. Im Monitor existiert wiederum ein Mechanismus, der die Informationen der einzelnen Zeilen entgegennimmt und sie auf dem Bildschirm abbildet. Wie funktioniert nun dieser Mechanismus? Der Einfachheit halber gehen wir zunächst von einer zweifarbigen (schwarz/weiß oder schwarz/grün) Darstellung aus.

In der Bildröhre des Monitors wird ein auf die Innenseite der Bildschirmoberfläche gerichteter Elektronenstrahl erzeugt. Durch sogenannte elektromagnetische Ablenkspulen

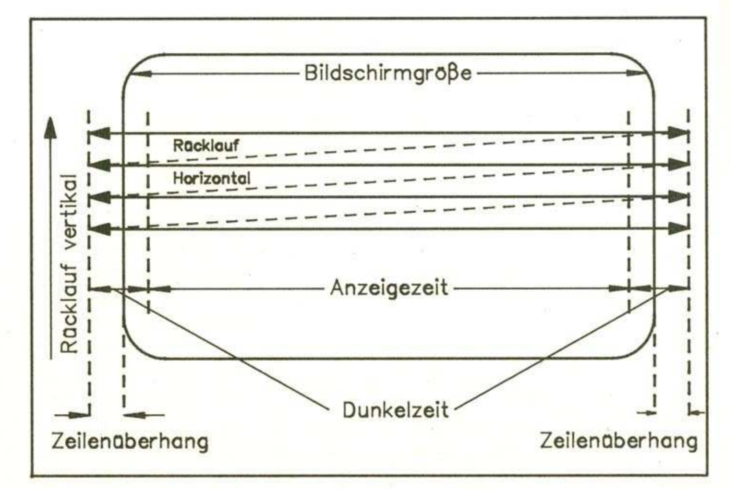


Bild 3. Das Zeilensprungverfahren schematisch dargestellt

Anzahl Bitplanes	verfügbare Farben
1	2
2	4
3	8
4	16
5	32

Tabelle. Die Abhängigkeit zwischen den Bitplanes und den verfügbaren Farben

läßt sich der Elektronenstrahl auf beliebige Punkte des Bildschirmes lenken. Trifft er auf die Schicht an der Innenseite der Bildschirmoberfläche, so wird diese zum Leuchten angeregt. Je nach der Intensität des Elektronenstrahls ist der Leuchteffekt mehr oder weniger intensiv. Auf einem Schwarzweiß-Fernseher werden auf diese Weise alle Graustufen einschließlich Schwarz und Weiß erzeugt.

Wie bereits erwähnt, schickt Denise seine Daten zeilenweise an den Bildschirm. Die Monitorelektronik lenkt daher den Elektronenstrahl, beginnend von der linken oberen Ecke, ebenfalls zeilenweise über den Bildschirm. (Anmerkung: Aus didaktischen Gründen haben wir hier Ursache und Wirkung vertauscht. Monitore beziehungsweise Fernseher haben schon immer zeilenweise gearbeitet. Denise hat sich danach zu richten.) Erreicht der Elektronenstrahl die rechte Schirmseite, so wird der Strahl dunkelgetastet (niedrigste Intensität) und rasch zur anderen Schirmseite zurückgeführt, wobei er eine Zeile tiefer gesetzt wird. Man nennt dies (horizontalen) Strahlenrücklauf (Bild 3). Die horizontale Weglänge des Strahls ist üblicherweise länger als die Bildschirmgröße. Der Betrag, um den der Elektronenstrahl die Schirmgrenzen überschreitet, wird als Zeilenüberhang (overscan) bezeichnet. Nach Abbildung der letzten Zeile springt der Strahl wieder in die linke obere Ecke und der beschriebene Vorgang beginnt von neuem.

Um die 625 Zeilen eines Fernsehbildes möglichst flackerfrei darzustellen, bildet der Elektronenstrahl zunächst die ungeraden Zeilen des Bildes (1,3,5..) und dann die geraden Zeilen (2,4..) auf dem Bildschirm ab. Da ein Strahldurchlauf vom oberen zum unteren Rand des Bildschirms trotzdem $\frac{1}{50}$ Sekunde dauert, kann ein Fernseher nur 50 Halbbilder und damit 25 Bilder pro Sekunde erzeugen. Für Computermonitore ist die Bildinformation bei den Halbbildern in der Regel gleich und somit sind nur bis zu 312 Zeilen pro Bild darstellbar. Ein Farbmonitor enthält nicht nur einen Elektronenstrahl, sondern gleich drei Stück davon. Auf der Mattscheibe befinden sich pro Bildpunkt drei kleine verschiedenfarbige (rot, grün, blau) Punkte. Jeder der drei Elektronenstrahlen zielt auf jeweils einen der Punkte. Man kann sich das etwa vorstellen wie drei Taschenlampen, die auf ein Fenster leuchten, in dem selbst wiederum drei Glasscheiben in den Farben Rot, Grün, und Blau eingesetzt sind. Werden alle Farben mit gleicher Intensität durchleutet, entsteht die Mischfabe Weiß. Ändert man die Intensität der einzelnen Lampen, erhält man die entsprechenden Grund- beziehungsweise Mischfarben. Ansonsten arbeitet auch der Farbmonitor wie sein schwarzweißer Kollege im Zeilensprungverfahren.

Wie wir bereits beschrieben haben, gibt es für jeden Punkt auf dem Bildschirm eine Speicherstelle, in der Denise die Information findet, ob dieser Punkt auf dem Bildschirm gesetzt werden soll oder nicht.

Eine Landkarte im Speicher

Um der Monitorelektronik nun noch eine Farbinformation zu liefern, benötigt Denise weitere Daten. Dazu wird ein weiterer Speicherbereich angelegt, der genauso groß ist wie der bisherige. Nun stehen für jeden Punkt zwei Bit zur Verfügung. Beide Bits können entweder 0 oder 1 sein. Das ergibt vier unterschiedliche Kombinationen: 00 01 10 11. Verbinden wir nun jede dieser Kombinationen mit einer Farbe, so kann Denise mit zwei Bitplanes jedem Punkt eine von vier Farben zuordnen. Durch Kombinieren können Sie nun herausfinden, wieviel Kombinationen und damit Farben mit drei, vier oder fünf Bitplanes realisierbar sind. Ersparen wir uns dies durch eine einfache Rechnung am Beispiel von drei Bitplanes:

Jede der Bitplanes hat eine bestimmte Wertigkeit, mit dem der jeweilige Bitwert multipliziert wird. Bitplane 1 hat die Wertigkeit 1, Bitplane 2 die Wertigkeit 2 und Bitplane 3 schließlich die Wertigkeit 4. Sind die Bits aller drei Planes für einen bestimmten Punkt 0, so ergibt sich durch die Berechnung $0*1+0*2+0*3$ der

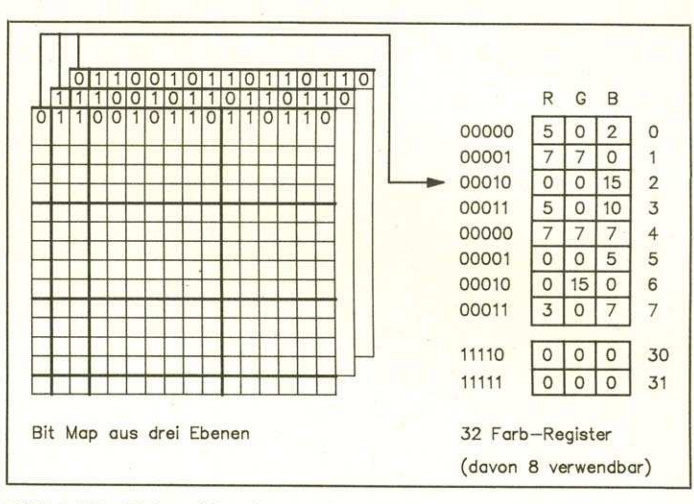


Bild 4. Die Bitkombinationen der Bitplanes ergeben die Farbe

Wert 0 und damit die Farbnummer 0. Sind alle Bits der Planes 1, so ergibt die Berechnung $1*1+1*2+1*4$ das Ergebnis 7. Alle anderen Kombinationen können nur Werte zwischen 0 und 7 ergeben. Somit erhalten wir die Farbnummer 0 bis 7 — das sind insgesamt acht Farben. Wie sie gesehen haben, verdoppelt sich die Anzahl der Kombinationsmöglichkeiten durch Hinzufügen einer Bitplane. Mit der Hinzunahme einer vierten beziehungsweise fünften Bitplane (Wertigkeit 8 und 16) erzielen wir also 16 beziehungsweise 32 verschiedene Farbnummern (Tabelle).

Bisher haben wir durch die Anzahl der Bitplanes lediglich festgelegt, wieviel unterschiedliche Farben auf dem Bildschirm abgebildet werden können. Es fehlt noch die Information, welche Farben das nun eigentlich sind.

Zu diesem Zweck existiert für jede der ermittelten Farbnummern eine bestimmte Speicherstelle (Register genannt). Diese sogenannten Farbregister — es gibt insgesamt 32 davon — bestimmen die Farbe des Punktes. Hierzu werden die 16 Bit, die ein solches Register enthält, in vier 4-Bit-Gruppen zerlegt. Jeweils drei dieser Gruppen bestimmen mit ihren vier Bit den Rot-, Grün- und Blauanteil der Farbe. Die vierte Gruppe bleibt ungenutzt.

Jede beliebige Farbe läßt sich ja aus diesen drei Grundfarben erzeugen. Mit den Bits einer jeden 4-Bit-Gruppe sind 16 verschiedene Kombinationen aus Nullen und Einsen möglich. Damit lassen sich durch die 16 Variationen eines Farbanteils $16*16*16=4096$ verschiedene Farben erzeugen. Durch die Einschränkung der Hardware auf fünf Bitplanes können in der Regel höchstens 32 Farben aus dieser großen Farbpalette ausgewählt werden. Durch spezielle Techniken, die wir in späteren Abschnitten noch kennenlernen werden, kann diese Beschränkung umgangen und bis zu 4096 Farben gleichzeitig dargestellt werden.

Bisher haben wir die Punktdichte von 320 x 200 Punkten als »normale« Auflösung bezeichnet. Das ist nicht ganz korrekt. Bedingt durch die unterschiedlichen Fernsehnormen hat der europäische PAL-Amiga ab der Betriebssystemversion 1.2 eine vertikale Auflösung von 256 Punkten, während sein amerikanischer Kollege mit dem NTSC-System dort nur 200 Punkte darstellen kann. Die horizontale Punktdichte ist bei beiden Computern gleich und beträgt 320 beziehungsweise 640 Punkte.

Für eine genauere Darstellung besitzt der Amiga einen hochauflösenden Modus. Hierbei verdoppelt sich die horizontale Anzahl der verfügbaren Punkte auf 640. Verständlicherweise benötigt der Computer für die hohe Auflösung doppelt soviel Speicherplatz. Da mit der derzeitigen Speichertechnologie die Daten jedoch nicht schnell genug aus dem Speicher geholt werden können, ist in diesem Modus die Anzahl der Bitplanes auf vier und die der verfügbaren Farben damit auf 16 eingeschränkt.

Wahrscheinlich haben Sie schon einmal etwas über die weiteren Grafikmodi Ihres Computers gehört: dem Interlace- und Hold & Modify-Modus. In der nächsten Folge dieses Kurses zeigen wir Ihnen, wie diese funktionieren und der Programmierer Einfluß nehmen kann auf die Grafikdarstellung des Amiga.

(Joschy Polierer/pa)



C-Kurs für Anfänger

In diesem zweiten Teil des C-Kurses für Anfänger wollen wir uns mit dem großen Gebiet der Daten beschäftigen. Denn ohne zu verarbeitende Daten kann natürlich das beste Programm nichts bewirken.

Daten können grob in Konstanten und Variablen unterteilt werden. Die Unterschiede und die Verwendung werden Ihnen heute klarwerden. Zuerst werden wir uns mit den Konstanten unter C befassen.

Der Wert einer Konstanten ist festgelegt und nicht veränderbar. Das heißt, daß Konstanten immer dann Anwendung finden, wenn tatsächlich unveränderliche Werte benötigt werden. Ein Beispiel wäre die Erdbeschleunigung, die den Wert 9,81 besitzt. Die Mehrwertsteuer sollte man nicht als Konstante benutzen, da sie sich ändern kann und ein Umschreiben des Programms erforderlich machen würde.

Bei den Konstanten unterscheidet man:
— ganzzahlige Konstanten
— reelle Konstanten
— Zeichenkonstanten
— Zeichenkettenkonstanten

Bei den ganzzahligen Konstanten müssen wir nach
— dezimalen ganzzahligen Konstanten
— oktalen ganzzahligen Konstanten
— hexadezimalen ganzzahligen Konstanten
unterscheiden (siehe Bild 1).

Dezimale Konstanten bestehen aus den Ziffern 0 bis 9. Die erste Ziffer einer dezimalen Konstanten darf niemals die Null sein, denn mit einer führenden Null werden Oktalzahlen gekennzeichnet. Beispiele hierfür sind:

```
10
-111
5423
```

Die Werte für ganzzahlige Konstanten reichen von -32768 bis 32767, da der Computer intern 16 Bit zur Darstellung verwendet. An dieser Stelle können wir uns die Formatangaben, welche im vorigen Kursteil behandelt wurden, ins Gedächtnis zurückrufen:

```
printf("Dezimalzahl
einhundert = %d\n",100);
```

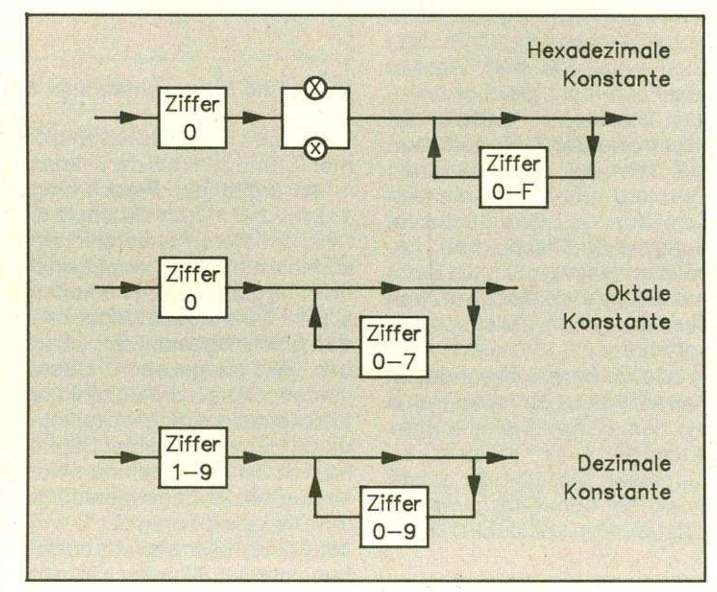


Bild 1. Ganzzahlige Konstanten der verschiedenen Zahlensysteme (hexadezimal, oktal und dezimal) als Syntaxdiagramm

ergibt die Ausgabe »Dezimalzahl einhundert = 100«. Wie wir im letzten Kursteil schon gesehen haben, steht »%d« als Platzhalter für eine Dezimalzahl. Die »100« ist der Wert, der dann an dieser Stelle ausgegeben wird. Sollen ganzzahlige Konstanten als long-Typen (32-Bit-Darstellung), das heißt als großer ganzzahliger Wert (Bereich von -2147483648 bis 214783647), gekennzeichnet werden, so muß an das Ende der Konstanten ein 'L' oder 'l' angehängt werden:

```
50000L    oder    50000l
```

Oktale Konstanten beinhalten die Ziffern 0 bis 7 und müssen mit einer Null beginnen. Einige Beispiele sind:

```
012
04222
0667
```

Da die Ziffer 8 im Oktalsystem nicht enthalten ist, sind folgende Oktalzahlen nicht erlaubt:

```
0282
080
```

Auch hier ein Beispiel mit printf():

```
printf("Oktalzahl 100 =
dezimal %d\n",0100);
```

Dieser Aufruf ergibt die Ausgabe »Oktalzahl 100 = dezimal 64«. Solche Ausgaben funktionieren auch in der anderen Richtung:

```
printf("Dezimalzahl 64 =
oktal %o",64);
     /* %o Formatangabe
für Oktalzahl */
```

Dieser Aufruf erzeugt die Ausgabe: »Dezimalzahl 64 = oktal 100«.

Hexadezimale Konstanten können aus 16 verschiedenen Ziffern bestehen. 0 bis 9 und die Zeichen A bis F. Hexadezimalkonstanten beginnen mit 0X oder 0x:

```
0xff
0X45
0xCAFE
```

Ein Beispiel soll das Ganze verdeutlichen:

```
printf("Hex-Zahl 100 =
dezimal %d\n",0X100);
```

Durch diesen Aufruf erhält man die Ausgabe »Hex-Zahl 100 = dezimal 256«.
Auch hier noch die Umkehrung:

```
printf("Dezimal-
zahl 256 = Hexadezimal
%x",256);
```

Die erzeugte Ausgabe hierzu: »Dezimalzahl 256 = Hexadezimal 100«

Die verwendeten Konstanten sind bis jetzt alle ganzzahlig gewesen, doch nun wenden wir uns den Real- oder Fließkommakonstanten (in C float) zu. Realkonstanten stellen Dezimalbrüche dar. Anstelle eines Kommas wird jedoch der Dezimalpunkt geschrieben. Sehr große oder sehr kleine Zahlen können in Potenzschreibweise dargestellt werden. Ein Schema dafür sehen Sie im Bild 2.

Einige Beispiele machen dies noch klarer:

```
 876.123
-66.987659
 123.11E2
 123.11e-2
```

Hier eine interessante Erweiterung der Formatangabe für eine Fließkommazahl:

```
printf("PI auf 2 Stellen
genau: %1.2f",3.141592);
```

ergibt die Ausgabe »PI auf 2 Stellen genau: 3.14«.

Die Angabe '%1.2f' besagt, daß mindestens eine Stelle vor

Teil 2

KURSÜBERSICHT

Mit diesem Kurs sollen Sie auf dem Amiga die Grundzüge der C-Programmierung erlernen. Kenntnisse anderer Programmiersprachen sind nützlich, aber nicht nötig.

TEIL 1: Allgemeine Einführung und Programmaufbau mit Beispielprogramm.

TEIL 2: Daten, aufgeteilt nach Konstanten und Variablen

TEIL 3: Zusammengesetzte Datentypen, Zeiger, Typ-Konvertierung und Operatoren
TEIL 4: Ausdrücke und Anweisungen
TEIL 5: Funktionen und deren Aufruf
TEIL 6: Der C-Präprozessor und die Bibliotheksfunktionen

und zwei Stellen nach dem Punkt dargestellt werden sollen. Die Angabe »%4.0« würde bedeuten, daß wenigstens vier Stellen vor und keine Stelle nach dem Punkt erscheinen sollen.

»%3f« verlangt eine Float-Zahl, die mindestens drei Zeichen lang ist. Eine Zahl mit zwei Stellen nach dem Dezimalpunkt erhält man mit »%.2f«. Eine Mindestbreite ist nicht festgelegt.

Wenn übrigens nach dem Prozentzeichen ein Minuszeichen folgt, wird das Ausgabefeld linksbündig dargestellt.

Als Besitzer des Aztec C-Compilers müssen Sie beachten, daß beim Linken die Bibliothek »m32.lib« mit eingebunden werden muß, wenn ein Programm mit float-Zahlen arbeitet. Diese Library muß vor der »c32.lib« aufgerufen werden. Beim Lattice-C-Compiler gilt dasselbe, nur heißt die Bibliothek hier »m.lib«.

Soweit zu den Zahlenkonstanten, mit denen Sie noch etwas üben sollten. Unser nächstes Thema sind Zeichenkonstanten.

Eine Zeichenkonstante ist ein einzelnes Zeichen, welches von zwei Anführungsstrichen umgeben ist. Dieses Zeichen kann ein Buchstabe, eine Ziffer oder ein Sonderzeichen sein:

```
'A'
'x'
'\n'
'\033'
```

Jedes Zeichen hat einen Zahlenwert, der in einer Tabelle festgelegt ist. Beim Amiga ist dies die ASCII-Tabelle, die Sie im Anhang Ihres Basic-Handbuches finden. Den Syntaxgrafen für die Zeichenkonstanten sehen Sie in Bild 3.

Folgender Aufruf

```
printf("%d ist der
ASCII-Wert von '%c'\n",
'A','A');
```

ergibt die Ausgabe »65 ist der ASCII-Wert von 'A'«.

Eine Zeichenkette ist eine Reihe beliebiger Zeichen, umgeben von Anführungszeichen(»"«). Zeichenketten werden auch als String bezeichnet.

```
"Das erste C-Programm"
```

ist eine Zeichenkette. Sie besteht aus 21 Zeichen. Sie zählen nur 20 Zeichen? Das ist auch richtig: In C wird jeder Zeichenkette noch ein Nullbyte (\0) angehängt, um deren Ende zu kennzeichnen. Unsere Zeichenkette wurde also folgendermaßen im Speicher abgelegt:

```
Das erste C-Programm\0
```

Da das Nullbyte ein Steuerzeichen ist, können Sie es nicht sehen. Die Ausgabe einer Zeichenkette mit printf geschieht wie bei Zahlen.

```
printf("%s","Zeichenket-
te")
```

Die Möglichkeit, eine Zeichenkette mit der Formatangabe "%s" (für string) auszugeben, gewinnt erst bei der Arbeit mit Variablen an Bedeutung.

Beachten Sie, daß "\n" nicht gleich '\n' und auch "a" ungleich 'a' ist!

Den Beweis hierfür bringt die Formatangabe "%c" in der printf()-Funktion. Es wird versucht "a" und 'a' als Zeichen (character) darzustellen.

```
100 main()
110 {
120      printf("Test 1:
%c\n","a");
130      printf("Test 2:
 %c",'a');
140 }
```

Bei dem Beispiel funktioniert die Ausgabe in Zeile 120 nicht, da versucht wird, eine Zeichenkette als Zeichen auszugeben. Auch wenn eine Zeichenkette nur einen Buchstaben enthält, ist sie kein Zeichen.

Insgesamt haben Konstanten zwei entscheidende Nachteile:

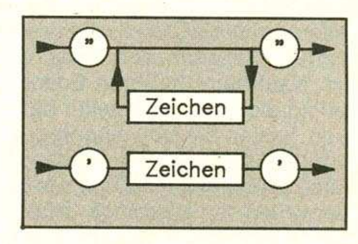


Bild 3. Die Syntaxdiagramme für Zeichen- und Zeichenkettenkonstanten gehören zu den einfachsten

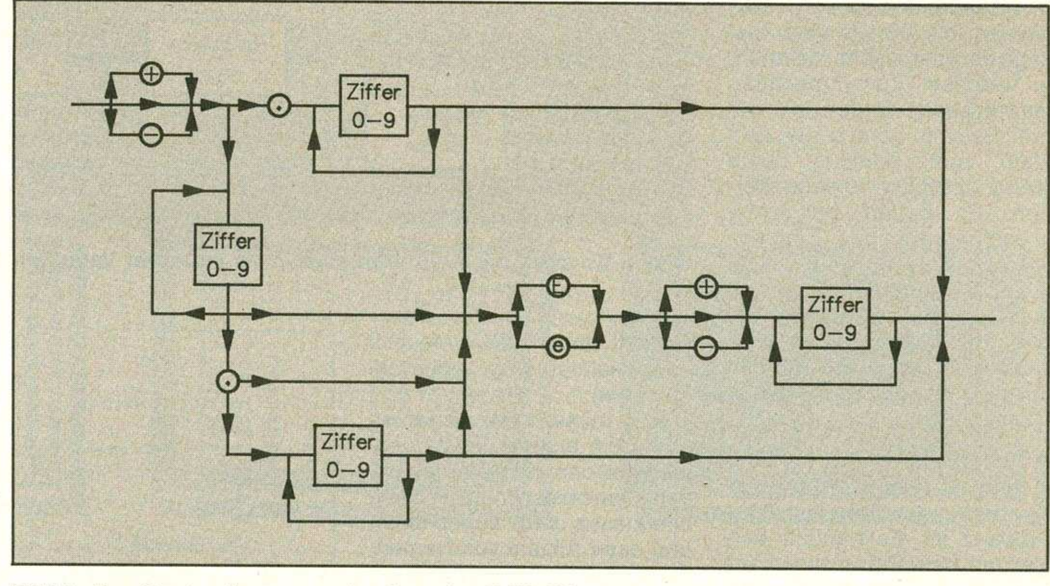


Bild 2. Das Syntaxdiagramm der Sprache C für Fließkommakonstanten ist sehr komplex. Wie es zu lesen ist, erfahren Sie am Ende des Kurses im Textkasten.

— Ein Programm, in dem viele geheimnisvolle Zahlen enthalten sind, ist nur sehr schwierig zu lesen
— noch schwieriger ist es jedoch, ein solches Programm zu ändern.
Beispiel:

Sie haben einen Editor geschrieben, der mangels Speicherplatz nur 1000 Textzeilen aufnehmen kann. In allen Abfragen, Entscheidungen und so weiter steht die Zahl 1000, da Sie ja den Bediener informieren müssen, wenn die maximale Zeilenanzahl erreicht ist. Nachdem Sie Ihren Editor erfolgreich fertiggestellt haben, leisten Sie sich eine Speichererweiterung. Sie könnten jetzt problemlos 5000 Zeilen verwalten — allerdings müssten Sie dafür alle in Frage kommenden Konstanten »1000« in Ihrem Quellprogramm durch »5000« ersetzen. Eine sehr mühselige Arbeit. Symbolische Konstanten schaffen Abhilfe. Anstelle von nichtssagenden Zahlen treten Namen, die bereits eine Aussage darüber machen können, in welchem Zusammenhang sie eingesetzt werden. In unserem Beispiel könnte die Zahl 1000 zum Beispiel durch »MAX_ZEILEN« ersetzt werden. Es bedarf nur einer #define-Anweisung. Eine solche #define-Anweisung muß außerhalb jeder Funktion und auch vor der ersten Anwendung der Konstanten definiert werden. In der Regel schreibt man sie an den Anfang des Quelltextes:

```
#define MAX_ZEILEN 1000
```

Beachten Sie, daß das erste Zeichen in der Zeile das Doppelkreuz ist. Es handelt sich hier um eine Präprozessoranweisung. Der Präprozessor ersetzt in unserem Beispiel bei jedem Auftauchen den Text »MAX_ZEILEN« durch den Text »1000«. Dies geschieht vor dem eigentlichen Übersetzen des Quelltextes. Zeilen, in denen symbolische Konstanten vereinbart werden, dürfen nicht mit einem Strichpunkt abgeschlossen werden, denn MAX_ZEILEN würde dann durch »1000;« ersetzt werden, was sich vielleicht nicht ganz mit Ihren Vorstellungen decken würde. Achten Sie hierauf besonders, denn solche Fehler sind nur sehr schwer zu finden.

Die Namen einer symbolischen Konstanten sollten in Großbuchstaben geschrieben werden, um sie von Variablennamen unterscheiden zu können. Wird hinter dem Namen einer #define-Anweisung kein Wert angegeben, so wird der Wert 1 angenommen. Eine Änderung des Wertes ist nicht möglich. Eine symbolische Konstante behält ihren Wert, bis sie mit einer weiteren Präprozessor-Anweisung als undefiniert erklärt wird:

```
#undef MAX_ZEILEN
```

Danach kann die symbolische Konstante wieder neu verwendet werden. Die Präprozessoranweisungen werden in einem späteren Kursteil noch ausführlicher behandelt.

Variablen sind Datenobjekte mit Namen, denen Werte zugewiesen werden können. Im Gegensatz zu symbolischen Konstanten sind jedoch die Variablenwerte, wie der Name schon sagt, variabel oder veränderbar. Man unterscheidet einfache und zusammengesetzte Variablen. In diesem Kursteil werden die einfachen oder skalaren Datentypen besprochen.

In C müssen alle Variablen vor ihrem ersten Einsatz definiert werden. Eine Variablendefinition besteht aus der Speicherklasse, dem Variablentyp und einer Anzahl von Namen, die diesem Typ angehören. Wie die Variablendefinition auszusehen hat, können Sie dem Syntaxgrafen in Bild 4 entnehmen.

Ein kleines Beispiel soll zur Verdeutlichung beitragen:

```
100 ...
110 void funktion()
120 {
130        static int
name1, name2;
140        ...
150 }
160 ...
```

In der Funktion »funktion()« wurden zwei Variablen des Typs »int« mit der Speicherklasse »static« vereinbart. Die Speicherklassen nehmen wir zuerst unter die Lupe, da sie für alle Variablentypen gelten. Die Programmiersprache C erlaubt die Verwendung von lokalen und globalen Variablen. Lokal bedeutet, daß eine Variable nur in der Funktion (oder dem Funktionsblock), in der (dem) sie definiert wurde, Gültigkeit besitzt. Soll jedoch eine Variable in der ganzen Quelldatei oder gar im gesamten Quellprogramm ansprechbar sein, so muß diese als globale Variable definiert werden. Globale Variablen stellen, neben der Parameterübergabe, eine Alternative zum Datenaustausch zwischen Funktionen dar.

Die oben beschriebenen Geltungsbereiche (lokal oder global) werden mit Hilfe von vier Speicherklassen realisiert:

- auto
- register
- extern
- static

Variablen, die den Speicherklassen »auto« und »register« angehören, sind lokale Variablen. Sie werden beim Aufruf der Funktion erzeugt und beim Verlassen wieder gelöscht. Ihr Wert geht also verloren. Beim erneuten Aufruf der Funktion wird die Variable automatisch neu angelegt. Man spricht deshalb auch von dynamischen Variablen.

Wenn bei einer Variablendefinition innerhalb eines Anweisungsblocks die Speicherklasse fehlt, so wird die Klasse »auto« angenommen. Variablen der Speicherklasse »register« verhalten sich wie die der Klasse »auto«, nur werden sie nicht im Arbeitsspeicher abgelegt, sondern es wird versucht, sie in den Registern der CPU unterzubringen. Die Programme benötigen somit weniger Speicher und werden schneller abgearbeitet. Die Geschwindigkeitssteigerung kommt daher, daß der Zugriff auf die Register rascher geschieht, als der auf den Speicher. Ist keines der Register frei, so unterscheiden sich die Speicherklassen auto und register nicht.

Alle Variablen, die der Speicherklasse »extern« angehören, sind im ganzen Quellprogramm bekannt (uneingeschränkt global). Dabei spielt es keine Rolle, aus wie vielen Dateien dieses besteht. Es gibt jedoch auch hier eine Ausnahme, die diese Regel bestätigt. Wird in einem Anweisungsblock eine lokale Variable mit dem Namen einer extern-Variable definiert, so dominiert in diesem Block (oder Funktion) die lokale Variable. Solche Doppelnamen sollte man jedoch tunlichst vermeiden, da sie das Programm undurchsichtig machen und oft zu Fehlern führen.

Die Definition von externen Variablen geschieht außerhalb jeder Funktion. Dabei darf keine Speicherklasse angegeben werden.

In Listing 1 in Zeile 100 wird die externe Variable »name«

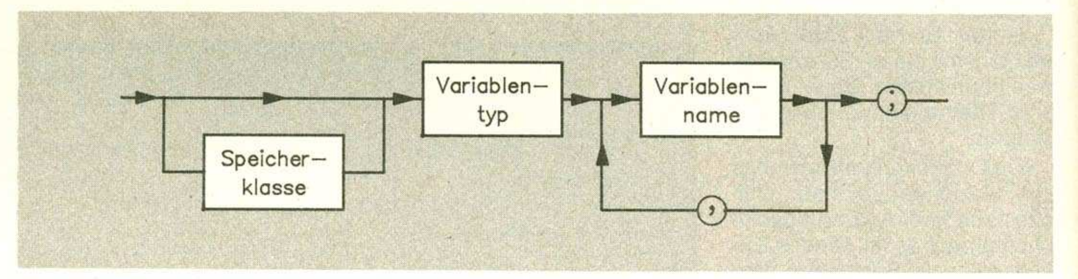


Bild 4. So sieht die Definition einer oder mehrerer Variablen schematisiert aus

```
100 int name;
110
120 void funktion()
130 {
140      /* Ausgabe: 'Der Wert von "name" ist 10.' */
150      printf("Der Wert von \"name\" ist %d.\n",name);
160 }
170
180 void main ()
190 {
200      name=10;          /* name erhält den Wert 10 */
210      funktion();       /* Funktionsaufruf */
220 }
```

Listing 1. Ein Beispielprogramm für externe Variablen

GO AMIGA!

GRATIS DEMONSTRATIONSPROGRAMME !

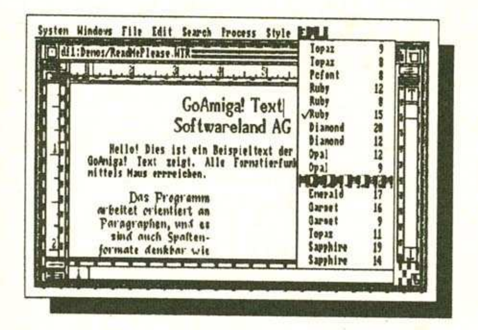
Sehr geehrter Kunde,

Damit Sie niemals die "Katze im Sack" kaufen, senden wir Ihnen zu unseren Produkten Gratisdemos, damit Sie sich selbst ein Urteil bilden. Zur problemlosen und schnellen Bearbeitung beachten Sie bitte folgendes:

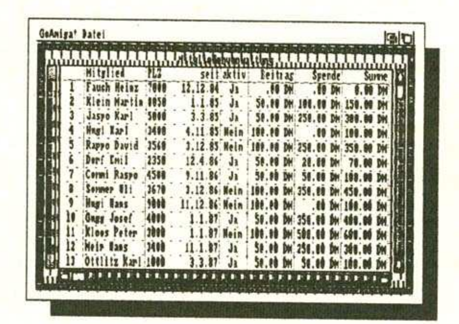
- Schicken Sie uns nur formatierte Disketten (3,5 Zoll), auf denen der Programmname und Ihre Adresse vermerkt sind.
- Ausserdem einen ausreichend frankierten Rueckumschlag (DM 0,80 reichen nicht), auf dem natuerlich auch Ihre Adresse steht.
- In Ihrem eigenen Interesse verwenden Sie bitte gepolsterte Umschlaege mit der Aufschrift "Datentraeger".

Herzlichst,

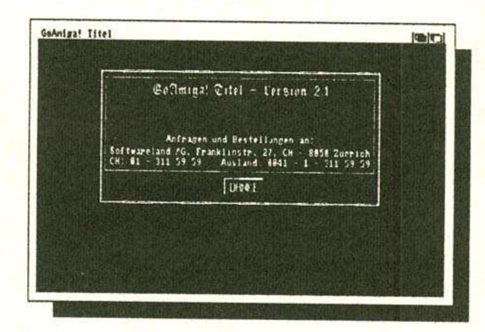
SOFTWARELAND AG ZUERICH



Die Textverarbeitung mit der besten Benutzerfuehrung. Datenaustausch mit GoAmiga! Datei moeglich. Spaltenorientiert. Rechtschreibkorrektor. Lieferbar ab ca. Januar 88. Deutsche Anleitung.
GOAMIGA TEXT DM 249.-



Die einzige Dateiverwaltung, die IFF-Bilder und IFF-Tonfolgen verarbeitet. Es erschienen mehr als fuenf begeisternde Testberichte. Multitasking. Deutsch-Englisches Woerterbuch. Deutsche Anleitung.
GOAMIGA DATEI DM 199.-



Der Titelgenerator. Das Programm erlaubt es, IFF-Grafiken, IFF-Tonfolgen und Laufschriften beliebig zu kombinieren. Erzeugt Filmvorspann auf allen Disketten. Deutsche Anleitung.
GOAMIGA TITEL DM 89.-

Bestellservice:

BRD: 0041-1-3115959
CH: 01-3115959

Geschäftszeiten:
10.00–12.30, 13.30–18.30 Uhr, außer montags,
Sa.: 10.00–16.00 Uhr.

softwareland
Franklinstraße 27
CH-8050 Zürich (Schweiz)

vom Typ »int« definiert. Zeile 200 beinhaltet ein neues Element der Sprache C: den Zuweisungsoperator »=«. Durch ihn erhält die Variable name den Wert 10. Beachten Sie jedoch, daß dies in der Funktion main() geschieht. In main() wurde keine Variable dieses Namens definiert und dennoch hat sie Gültigkeit! Auch in der Funktion »funktion()« kann name angesprochen werden.

Die Variable ist durch die externe Definition in Zeile 100 global bekannt. Durch den Funktionsaufruf in Zeile 150 wird der Wert der Variablen auf dem aktuellen Ausgabegerät, dem Bildschirm, ausgegeben.

Wie Sie sehen, wurde bei der Definition der Variablen die Speicherklasse »extern« nicht aufgeführt. Diese wird ausschließlich bei der Deklaration von Variablen oder Funktionen angegeben.

Variablen, die innerhalb einer Funktion mit der Speicherklasse »static« definiert wurden, sind zwar lokale Variablen, behalten aber, im Gegensatz zu den auto- und register-Variablen, ihren Wert bei.

Werden sie jedoch extern, also außerhalb einer Funktion vereinbart, so haben sie in der gesamten Quelldatei (nicht Quellprogramm !) Gültigkeit.

Beachten Sie, daß die unter »extern« und »static« genannten Regeln auch für Funktionsnamen gelten. Die Speicherklassen »auto« und »register« haben dagegen für Funktionen keine Bedeutung, da die Funktionsdefinition innerhalb einer Funktion in C nicht erlaubt ist.

Eine externe Variable behält von dem Punkt der Definition bis zum Ende des Quellprogramms ihre Gültigkeit. Wenn sie nun aber in der Quelldatei »A« eines Programms definiert ist und auch in der Datei »B« angesprochen werden soll, muß die Variable in der Datei »B« deklariert werden.

Der Unterschied zwischen Deklaration und Definition wird oft übersehen, ist aber von großer Bedeutung. Bei der Definition von Variablen werden Datentyp und Speicherbedarf festgelegt und der entsprechende Speicherplatz reserviert. Bei der Deklaration hingegen wird nur an den Datentyp und Speicherbedarf einer Variable »erinnert«.

Wichtig ist dies auch für die Verwendung einer externen Variablen vor ihrer Definition, sie muß nämlich vor dem ersten Erscheinen deklariert werden. In diesem Fall spricht man von einer Vorwärtsdeklaration. Bei der Deklaration einer Variablen wird die Speicherklasse »extern« angegeben. Diese Regeln haben ebenfalls für Funktionen Gültigkeit.

```
100 ...
110 extern int var;
120 extern void fun();
140 ...
```

Eine Deklaration von Variablen kann außerhalb, aber auch innerhalb einer Funktion stattfinden. Entsprechend ist ihr Geltungsbereich. Am Ende des letzten Kursteils haben wir begonnen, ein Syntaxdiagramm für ein Programm-Modul zusammenzustellen. Nachdem wir die Extern-Definition und die Deklaration kennengelernt haben, können wir dieses um zwei Punkte erweitern. Das Ergebnis sehen Sie im Syntaxgraph für das Modul (Bild 5).

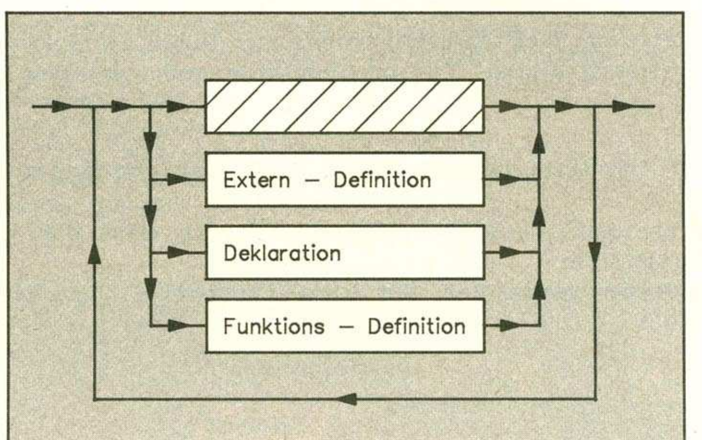


Bild 5. Das fast vollständige Syntaxdiagramm für ein Modul

Nun wollen wir weiter auf die einfachen oder skalaren Variablen eingehen (siehe Bild 6). Elementare Datentypen sind:

- int
- float
- char
- void

Hierbei betrifft void nur Funktionen und bedeutet, daß die Funktion keinen Rückgabewert liefert. Variablen vom Typ int können, wie bereits oben erwähnt, nur ganze Zahlen aufnehmen.

```
100 void main()
110 {
120         int ganze_zahl;
130         ganze_zahl=100;
140 }
```

In Zeile 120 wird die Variable »ganze_zahl« als Integer-Variable vereinbart. Das heißt, die Variable kann nur ganze Zahlen beinhalten. In Zeile 130 wird, mit Hilfe des Zuweisungsoperators (»=«), der Variablen der Wert 100 zugewiesen. Wenn bereits bei der Definition einer Variablen dieser ein Wert zugewiesen wird, spricht man von einer initialisierenden Definition. In diesem Fall ersetzt man die Zeile 120 durch

```
int ganze_zahl = 100;
```

Die Zeile 130 kann dann entfallen. Diese Methode empfiehlt sich, wenn Variablen mit einer Voreinstellung belegt werden sollen.

```
100 int dez_zahl;
110
120 void input()
130 {
140     printf("\n\n\tBitte geben Sie eine Dezimalzahl an: *\b");
150     scanf("%d",&dez_zahl); /*Die neue Funktion */
160 }
170
180 void main()
190 {
200     printf("\f\n\tUmrechnung DEZ -> HEX\n\t-------------------
    ");
210     input();        /* Funktionsaufruf */
220     printf("\n\n\tDEZ %d = HEX %x\n",dez_zahl,dez_zahl);
230 }
```

Listing 2. Zur Umwandlung von Dezimal- in Hexadezimalzahlen dient dieses kurze C-Programm

Aber jetzt noch eine kleine Anwendung, bei der wir unser Wissen unter Beweis stellen können: Eine Dezimalzahl soll über die Tastatur eingegeben und als Hexadezimalzahl ausgegeben werden.

Die Lösung (siehe Listing 2) hierzu ist nicht weiter schwierig, aber sie beinhaltet eine neue Funktion, die wir immer wieder benötigen werden.

Zeile 100 enthält die Variablendefinition der int-Variable »dez_zahl«. Sie ist eine externe Variable und hat somit in beiden Funktionen (main() und input()) Gültigkeit. In Zeile 200 wird die Funktion printf() aufgerufen. Der printf()-Befehl in Zeile 200 löscht den Bildschirm und gibt die Programmüberschrift aus und unterstreicht sie mit Minuszeichen. Anschließend erfolgt der Aufruf für die Funktion input(). In Zeile 140 wird der Text für die Eingabe ausgegeben. Das Zeichen »\b« (backspace) setzt den Cursor ein Zeichen nach links, über den Stern » * «.

Es bleibt nur noch das Problem, eine Dezimalzahl von der Tastatur zu lesen. Eigentlich ist dies gar kein Problem, denn wie printf() für die Ausgabe, so existiert auch eine Funktion, die für die Eingabe von Zeichen verantwortlich ist: scanf(). In Zeile 150 wird scanf() aufgerufen. Da diese Funktion in diesem Kurs noch nicht besprochen wurde, soll sie, bevor es mit den Variablentypen weitergeht, noch ein wenig beleuchtet werden:

```
scanf("Formatangabe",
Liste von Variablenzei-
gern);
```

Die Funktion scanf() ist sozusagen das Gegenstück zu printf(). Es werden Zeichen vom Standardeingabegerät, in der Regel der Tastatur, gelesen. Diese Zeichen werden, wie in der Formatangabe vorgeschrieben, umgewandelt und der Variablen, auf die der Variablenzeiger verweist, zugewiesen. Wichtig hierbei ist, daß scanf() einen Zeiger auf eine Variable verlangt. Dies ist nötig, damit das Progamm den eingegebenen Wert in die richtigen Speicherstellen schreiben kann. Eine Möglichkeit, einen Variablenzeiger zu erhalten, sehen Sie in Zeile 150 (&dez_zahl). Auf Zeiger wird später noch mit einem kleinen Programm eingegangen.

Auf das Prozentzeichen der Formatangabe dürfen folgen:
* — Das nächste Eingabefeld wird ignoriert
Dezimalzahl — Bestimmung der Länge des Eingabefeldes. Wird keine Länge angegeben, so wird die Eingabe bis zum nächsten Space, Tabulatorzeichen oder Return gelesen.
d, o, x, f, c, s - Dezimalzahl, Oktalzahl und so weiter, wie bei printf() bereits erklärt. Der Variablenzeiger muß auf eine Variable des entsprechenden Typs zeigen.

Starten Sie doch noch einmal unser kleines Programm »DEZ->HEX« und geben Sie diesmal eine sehr große Dezimalzahl an, zum Beispiel 50000. Sie werden feststellen, daß das Programm nicht mehr korrekt arbeitet. Der Grund hierfür ist der Wertebereich der benutzten Integer-Variablen (siehe Tabelle 1).

Um große Zahlen verarbeiten zu können, müssen wir also mit »long int«- oder kurz »long«- Variablen operieren. Ändern Sie hierzu in unserem Programm folgende Zeilen:

```
100 long dez_zahl; /*
Externe int-Variable */
150       scanf("%D",
&dez_zahl);
290       printf("\n\n\
tDEZ %ld = HEX %lx\n",
dez_zahl,dez_zahl);
```

Wenn die Variable, auf die der Zeiger in einer scanf()-Funktion verweist, vom Typ »long« oder »double« ist, so müssen die Buchstaben »d«, »o«, »x« oder »f« groß geschrieben werden. Bei der printf()-Funktion muß dagegen in solch einem Fall dem jeweiligen Buchstaben ein »l« vorangestellt werden:

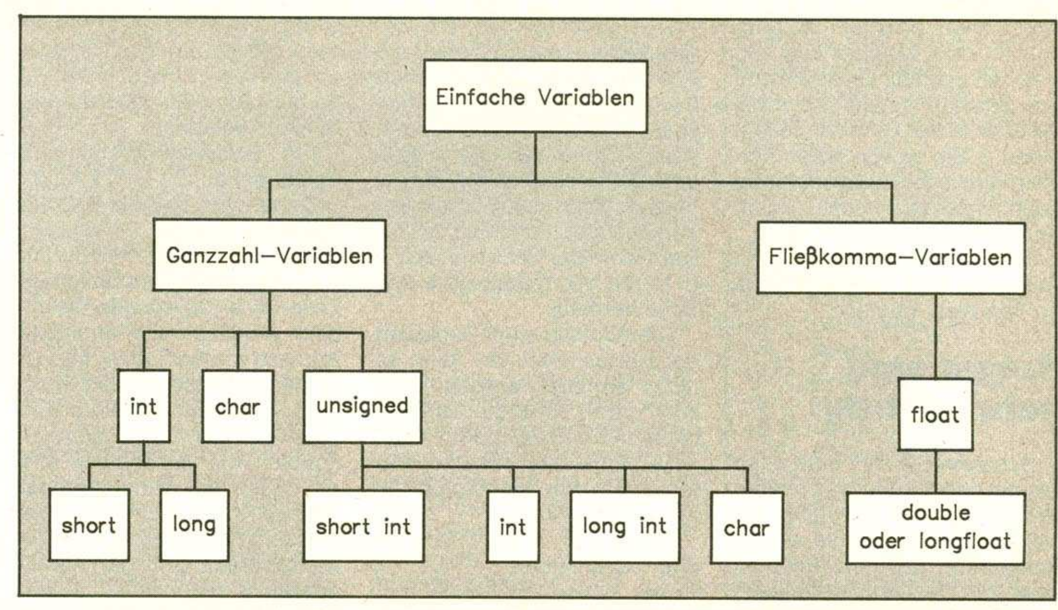


Bild 6. So teilen sich die einfachen Variablen in der Programmiersprache C auf

```
printf("Eine große Zahl:
%ld",50000);
```

Variablen des Typs »float« können reelle Werte einer bestimmten Genauigkeit aufnehmen. Die doppelte Genauigkeit bieten »long float«- oder »double«-Variablen.

```
100 ...
110 float a, b, c;
120 float pi = 3.1415;
140 double grosse_zahl;
150 ...
```

Je genauer eine Gleitkommzahl dargestellt werden soll, um so mehr Speicherplatz muß reserviert werden. Wieviel Speicher für eine Variable reserviert wird, ist abhängig von dem Compiler, den Sie verwenden. Die Größe einer float-Variablen auf Ihrem Compiler, in Bytes, zeigt Ihnen das folgende Programm:

```
100 /*
110 MEM: Speicherbedarf
von reelen Variablen
120 */
130
140 main()
150 {
160       printf("\n\
nSpeicherbedarf von
float-Variablen:\n\n");
170       printf("float
= %d Bytes\n",sizeof
(float));
180       printf("double =
%d Bytes\n\n",sizeof
(double));
190 }
```

Neu in diesem Programm ist nur der sizeof-Operator in Zeile 170 und 180. sizeof liefert die Größe seines in Klammern folgenden Operanden in Bytes. Dieser Wert wird, mit Hilfe der Formatanweisung »%d«, in den Ausgabetext eingefügt.

Zeichenvariablen (Character-Variablen) können die Werte von Zeichen aufnehmen.

Der Amiga arbeitet mit dem ASCII-Code, welcher aus 256 Zeichen besteht. Diese Anzahl von Zeichen lassen sich mit 8 Bit oder einem Byte darstellen. Denn 8 Bit, wovon jedes den Wert eins oder null annehmen kann, ergeben 2 hoch 8 = 256 Kombinationsmöglichkeiten.

Der sizeof-Operator, mit »char« als Operand, würde also den Wert 1 liefern.

Zeiger sind sehr wichtig

Schauen Sie sich bitte noch einmal den Quellcode unseres Programms DEZ->HEX an. Versuchen Sie ihn so umzuschreiben, daß ein Zeichen über die Tastatur eingelesen und dessen ASCII-Wert ausgegeben wird.

Unsigned ist kein selbständiger Datentyp, sondern findet nur in Verbindung mit den Typen short int, int, long int und char Verwendung. Unsigned bedeutet vorzeichenlos. Es können in C also vorzeichenlose Objekte, das heißt Objekte, die nur positive Werte oder den Wert null annehmen können, definiert werden.

```
100 ...
110 unsigned short int
    a, b, c;
120 unsigned int d;
130 unsigned long int
    e, f;
140 unsigned char g, h;
150 ...
```

Bisher haben wir Variablen immer mit ihrem Namen angesprochen und so ihren Wert erhalten. Die Programmiersprache C erlaubt darüber hinaus noch das Ansprechen einer Variablen über Zeiger (Pointer). Ein Zeiger enthält die Adresse einer Variablen. Im Zusammenhang mit der Funktion scanf() haben Sie bereits mit einem Zeiger gearbeitet, nämlich mit »&dez_zahl«.

Das Zeichen »&« ist der Adreßoperator und liefert die Adresse eines Objekts. Zum Beispiel muß die Funktion scanf() einen eingegebenen Wert in eine Variable schreiben. Dies ist jedoch erst durchführbar, wenn sie die Speicheradresse der Variablen kennt. Der Name der Variablen liefert ja deren Wert und nicht den Speicherplatz.

Die Adresse einer Variablen der Klasse »register« kann so nicht ermittelt werden. Das liegt daran, daß die Variable nicht im Speicher steht.

Mit Hilfe des Inhaltsoperators »*« lassen sich Zeiger direkt definieren:

```
int *int_zeiger
float *float_zeiger
char *char_zeiger
```

Wenn Sie die Definition von rechts nach links betrachten, liest man den Inhaltsoperator als »Zeiger auf« oder »Adresse von«. Somit besagt die erste Zeile: int_zeiger ist ein Zeiger auf eine int-Variable oder int_zeiger ist die Adresse einer int-Variablen.

Die Größe eines Zeigers ist unabhängig von dem Typ der Variable, auf die sie deutet. Es wird ja nur auf eine Speicherzelle gezeigt und der Prozessor des Amiga kann 16 MByte adressieren, was einer 24-Bit-Adresse entspricht.

Ein Beispielprogramm für Zeiger sehen Sie in Listing 3.

Dabei wird in Zeile 130 ein Zeiger auf eine int-Variable (zeiger) und in Zeile 140 die int-Variable »variable« definiert. In Zeile 160 erhält »zeiger« die mit dem Adreßoperator ermittelte Adresse von »variable«. Mit Hilfe des Inhaltsoperators kann man jetzt über den Zeiger zum Inhalt der Variablen gelangen (Zeile 170). Tatsächlich wird der Wert von »variable« über »*zeiger« ausgegeben. In Zeile 250 sehen Sie, daß sich der Inhaltsoperator und der Adreßoperator gegenseitig aufheben.

Wie Sie wissen, ist der Wertebereich einer Variablen, gleich welchem Typ sie angehört, genau festgelegt. Manchmal würde jedoch auch ein wesentlich kleinerer Bereich vollkommen ausreichen, zum Beispiel bei einer Integer-Variablen, die als Flag dient und nur den Wert null oder eins annehmen soll.

Im Jahre 1978 wurde in C ein Sprachmittel installiert, mit dem man sich seine eigenen Datentypen mit speziellem Wertebereich selbst erstellen kann — die Aufzählungstypen (enumerations type). Der Wertebereich wird durch die Aufzählung aller Elemente festgelegt. Wie so etwas aussieht, sehen Sie im Syntaxdiagramm in Bild 7.

Die Definition eines Aufzähllungstyps kann im Programm wie folgt aussehen:

```
enum datentyp{
        element_null,
        element_eins,
        element_zwei,
        element_drei
};
```

Beachten Sie, daß hinter dem letztem Aufzählungselement kein Komma folgt. Vergessen Sie hinter der geschlossenen Klammer das Semikolon nicht!

Hier wurde ein neuer Datentyp namens »datentyp« definiert. In geschweiften Klammern werden alle Elemente des Typs aufgeführt. Das erste Element erhält den Wert null, jedes weitere ist immer um eins größer als das vorherige. Variablen des Datentyps »datentyp« nehmen also in unserem Beispiel Werte von 0 bis 3 an.

Man kann jedoch, mit Hilfe des Zuweisungsoperators (»=«), jedem Element einen beliebigen ganzzahligen Wert zuweisen. Alle folgenden Werte werden wieder um eins inkrementiert, falls kein Wert angegeben ist.

```
100
110 enum datentyp{
120       element_null,
          /* Wert=0 */
130       element_eins,
          /* =1 */
140       element_zwei=10,
          /* =10 */
150       element_drei,
          /* =11 */
160 };
```

```
100
110 void main()
120 {
130       int *zeiger;
140       int variable=10;
150
160       zeiger = &variable;
170       printf("\nDer Wert der Variablen 'variable' = %d\n",*zeiger);
180
190       variable = *zeiger+1;
200       printf("'variable' um eins inkrementiert: %d\n",variable);
210       printf("Der Wert von '*zeiger': %d\n",*zeiger);
220
230       *zeiger=99;
240       printf("Der Wert von 'variable': %d\n",variable);
250       printf("Nochmals dieser Wert: %d\n\n",*(&variable));
260 }
270
```

Listing 3. Testprogramm für die Funktion von Zeigern

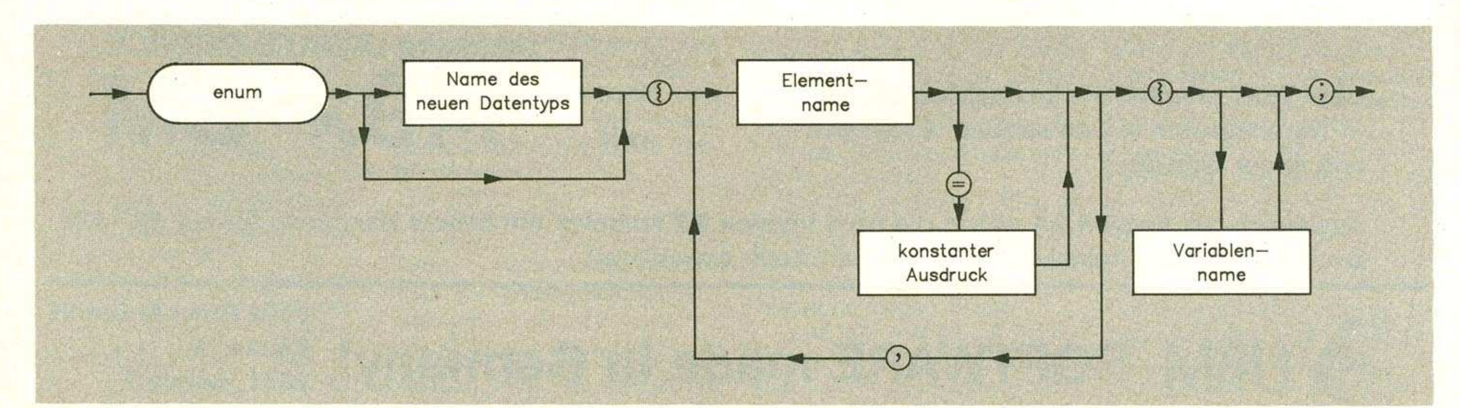


Bild 7. Der Aufzählungstyp (ein wichtiger Datentyp von C) als Syntaxdiagramm dargestellt

Nur an dieser Stelle darf einem Aufzählungselement ein Wert zugewiesen werden, denn ab der Definition gelten sie als Integer-Konstanten.

Nachdem wir jetzt einen neuen Datentyp vereinbart haben, müssen wir auch eine Variable dieses Typs definieren:

```
210 enum datentyp
     g_zahl1,g_zahl2;
```

Die Definiton des Datentyps und einer Variablen dieses Typs kann auch zusammengefaßt werden. Dazu ersetzt man Zeile 160 durch:

```
160 } zahl;
```

Außerdem darf der Datentyp auch ohne Namen definiert werden. Man kann in diesem Fall jedoch nur eine Variable dieses Typs definieren. Dies geschieht, indem Sie bei obigem Beispiel den Namen »datentyp« entfallen lassen. In jedem Fall kann die Variable »zahl« Werte zwischen »element_null« und »element_drei« erhalten.

Aufzählungstyp

Ein Beispielprogramm (siehe Listing 4), mit dem Sie etwas herumexperimentieren sollten, macht das Ganze klarer.

In den Zeilen 100 bis 160 werden ein Datentyp namens »steuerzeichen« und eine Variable »zeichen« dieses Typs vereinbart. Die Funktion »ausgabe()«, welche ein Zeichen auf dem Bildschirm ausgibt, wird in den Zeilen 180 bis 210 definiert. Danach wird »zeichen« immer ein anderer Wert zugewiesen und mit der Funktion ausgabe() auf dem Bildschirm ausgegeben.

Hiermit sind die skalaren Datentypen abgeschlossen. Schauen Sie sich alle Programme noch einmal genau an, experimentieren Sie mit ihnen.

Für die Spezialisten unter Ihnen haben wir noch zwei Nüsse zu knacken:

— lesen Sie zwei Zahlen über die Tastatur ein und geben Sie als Ergebnis die Summe auf dem Bildschirm aus. Entwickeln Sie verschiedene Lösungen. Vergleichen Sie die Programmlängen und die Längen der Quellcodes der verschiedenen Lösungen miteinander.

— schreiben Sie ein Programm, das eine ganze Zahl über die Tastatur einliest. Geben Sie die eingelesene Zahl als Zeichen aus, um sie als Steuerzeichen zu testen.

In unserem heutigen Kursteil haben Sie alles gelernt, was man über Konstanten und einfache Variable wissen muß. Vor allem den Teil mit den Variablen sollten Sie gut verstanden haben. Im nächsten Teil folgen die zusammengesetzten Variablen, die zum Großteil auf den einfachen aufbauen. Aber Sie können jetzt schon Werte über die Tastatur eingeben, sie verarbeiten und auf dem Bildschirm ausgeben. Versuchen Sie sich ruhig an eigenen Programmen, denn daraus lernt man am besten.

(Arno Gölzer/rb)

Typ	von	bis	Bit
char	-128	bis 127 =	8 Bit
short int	-32768	bis +32767 =	16 Bit
int	-32768	bis +32767 =	16 Bit
long int	-2147483648	bis +2147483647 =	32 Bit
unsigned char	0	bis 255 =	8 Bit
unsigned short	0	bis 65535 =	16 Bit
unsigned int	0	bis 65535 =	16 Bit
unsigned long int	0	bis 4294967295 =	32 Bit

Tabelle 1. Wertebereich der einfachen Variablen in C

```
100 enum steuerzeichen{
110     blitz = 7,
120     cls  =12,
130     ruecklauf,
140     zeichenI,
150     zeichenII
160 }zeichen;
170
180 void ausgabe()
190 {
200     printf("%c",zeichen);        /* Ein Zeichen ausgeben */
210 }
220
230 void main()
240 {
250
260     zeichen=cls;    /* Bildschirm löschen */
270     ausgabe();
280     zeichen=blitz;  /* Bildschirmblitz */
290     ausgabe();
300     zeichen=zeichenI; /* alternativ Zeichensatz einschalten */
310     ausgabe();
320     printf("\n\nDer andere Zeichensatz !!!"); /* Textausgabe
       ohne Return */
330     zeichen=zeichenII; /* normaler Zeichensatz einschalten */
350     ausgabe();
360     zeichen=ruecklauf; /* Cursor in die erste Spalte der
        aktuellen Zeile */
370     ausgabe();
380     printf("\t\t\t\tDer normale Zeichensatz !!!\n\n");
        /* Textausgabe */
390
400 }
```

Listing 4. Ein C-Programm mit dem Aufzählungstyp

Alle Quellcodes in diesem Kursteil werden mit den folgenden Anweisungen übersetzt:
Lattice-C V3.10:

```
lc Programmname
blink lib:c.o,Programmname.o lib lib:lcm.lib,
lib:lc.lib,lib:amiga.lib
```

Für Aztec-C V3.40 lauten die Aufrufe:

```
cc Programmname +l
ln Programmname.o -lm32 -lc32
```

Die Zeilennummern dürfen beim Abtippen natürlich nicht mit eingegeben werden.

Im letzten Kursteil haben wir über die Escape-Sequenzen gesprochen. Eingeleitet werden sie in printf-Befehlen mit »\033[«. Ein Beispiel wäre:

print("\033[2mTest")

Es können zum Beispiel folgende Codes eingesetzt werden:

Code	Bedeutung
0m	Normale Schrift
2m	Schwarz
3m	Kursiv
4m	Unterstrichen
7m	Revers
1A	Cursor hoch
1B	Cursor runter
1C	Cursor rechts
1D	Cursor links
1E	Cursor an den Anfang der nächsten Zeile
1F	Cursor an den Anfang der vorherigen Zeile
2x;yH	x und y sind Zeilen- und Spaltenpositionen, an die der Cursor gesetzt wird
1J	Löscht Bildschirm ab Cursor
1K	Löscht Zeile ab Cursor
1L	Fügt eine Zeile ein
1M	Löscht eine Zeile
1S	Hochscrollen
1T	Runterscrollen

Das Lesen von Syntaxdiagrammen:

Die in unserem Kurs verwendeten Syntaxdiagramme erleichtern es Ihnen, die Syntax von C zu verstehen. Dazu müssen Sie jedoch einiges wissen. In Kästen mit abgerundeten Ecken oder in Kreisen stehen nicht weiter unterteilbare Sprachelemente. Dies sind zum Beispiel die Buchstaben und Ziffern. In rechteckigen Kästen stehen Elemente, die durch die oben genannten ersetzt werden können. Zum Beispiel kann man in Bild 1 den Kasten mit dem Text »Ziffer 0-9« durch das Syntaxdiagramm für Dezimalziffern ersetzen.

Das Lesen dieser Diagramme ist dann recht einfach. Durch die Pfeile wird die Richtung angegeben, in der man weitergehen darf. Trennt sich ein Weg auf, können alle Richtungen benutzt werden. Dies ist in Bild 1 bei den hexadezimalen Konstanten zu sehen. Hier kann entweder »x« oder »X« verwendet werden. Im rechten Teil des Syntaxdiagramms ist noch zu sehen, daß Teile auch wiederholt werden dürfen. Die »Ziffer 0-F« kann durch die Schleife auch öfter benutzt werden.



Amiga 500-Buch

Das »Amiga 500-Buch« läßt sich grob in drei Teile gliedern. Im ersten Teil des Buches beschreibt der Autor die wesentlichen Grundzüge der Bedienung des Amiga. Durch diese Anleitung und praktische Übungen wird der Leser vertraut gemacht mit der grafisch orientierten Bedienung, der Anwendung der Maus und allen wichtigen Handgriffen, die er später tagtäglich benötigt. Die einfache Bedienung des Amiga ist im wesentlichen eine Leistung der Workbench. Und so beschäftigt sich der erste Teil auch ausführlich mit diesem Programm.

Der zweite Buch-Teil befaßt sich mit einer Art, den Amiga 500 zu bedienen, die nicht ganz so bildlich und intuitiv zu begreifen ist wie die Workbench: dem sogenannten CLI. Diese zweite Benutzerschnittstelle des Computers hat trotz der umständlicheren Handhabung ihre Vorteile. Sie wird meist von den Programmierern bei intensiver Nutzung der Betriebssystemkommandos verwendet.

Der dritte und letzte Teil wendet sich schließlich an diejenigen Leser, die neben Grundlagenwissen tiefer in die Materie einsteigen wollen. Der Autor beschreibt die Hardware des Amiga, was man damit machen kann sowie die Kontakte zur Außenwelt (Anschlüsse für Drucker und andere Zusatzgeräte). Die Programmierer unter den Lesern lernen — und solche die es noch werden wollen — anhand von ein paar kleinen Beispielprogrammen in Amiga-Basic, was man alles schon mit dieser einfachen und leicht erlernbaren Sprache auf dem Amiga anfangen kann. (Joschy Polierer/pa)

Markus Breuer, Amiga 500-Buch, Markt & Technik Verlag AG, 490 Seiten, ISBN 3-89090-522-6, Preis: 49 Mark

Amiga C in Beispielen

Dieses Buch ist eine grundlegende Einführung in C, die sich allerdings nicht an den absoluten Programmieranfänger, sondern eher an den Umsteiger von anderen Programmiersprachen wendet. Die Autoren folgen mit dem Aufbau des Buches amerikanischen Vorbildern, die nach einem einführenden Kapitel — »getting started« (oder: los geht's) genannt — den systematischen Teil enthalten. Diese Einführung ist für die Leser gedacht, die schnell zu einem laufenden C-Programm kommen wollen, um mit dem dabei gewonnenen Überblick die restlichen Kapitel als Handbuch zu nutzen.

Nach der Einführung beginnen die Autoren mit der unvermeidlichen Darstellung der »Daten und Datentypen«. Daran anschließend werden dem Leser mit den Operatoren und Befehlen zwei weitere und wichtige Elemente einer Programmiersprache vorgestellt. Die Ausführungen zu diesen Themen sind zwar knapp gehalten aber dennoch vollkommen ausreichend. Die Autoren verstehen es, einen derart trockenen Stoff verständlich aufzubereiten.



Folgerichtig befaßt sich das nächste Kapitel mit den im Befehlsumfang fehlenden Ein- und Ausgabebefehlen der C-Bibliothek. In einem späteren Kapitel gehen die Autoren dann noch einmal auf die grundlegende Arbeitsweise mit dieser Bibliothek ein. Diese Vorgehensweise liegt ganz im Sinne derjenigen Anwender, die möglichst schnell und ohne großen Leseaufwand Programme schreiben wollen.

In den weiteren Kapiteln werden die schon etwas komplizierteren Themen wie Arrays und Pointer (Zeiger), Strukturen und Verbunde behandelt. Ausführungen über Prozeduren — die »Unterprogramme« von C — sowie eine Beschreibung der C-Bibliothek und des Präprozessors runden das Bild der Sprache C ab.

Die Struktur von C verführt den Programmierer leicht dazu, komprimierte und damit unübersichtliche Programme zu schreiben. Im letzten Teil des Werkes befassen sich die Autoren deshalb mit dem Thema »Programmierstil in C« und der Programmoptimierung. Eine kleine Programmsammlung (sämtliche Programmbeispiele befinden sich auf einer mit dem Buch gelieferten Diskette) schließt diesen Teil und damit das Buch ab.

Natürlich ist »Amiga C in Beispielen« nicht das erste Grundlagenbuch für die C-Programmierer. Als solches kann es auch nur wenig auf die Systemprogrammierung des Amiga eingehen. Der Leser lernt aber Grundsätzliches über die Bedienung von C-Compiler, Linker und die Nutzung der Systembibliotheken auf dem Amiga. Dieses Wissen können Bücher, die sich nur allgemein mit C auseinandersetzen, nicht vermitteln. (Ingolf Krüger/pa)

E. Huckert/Frank Kremser, Amiga C in Beispielen, Markt & Technik Verlag AG, 237 Seiten, ISBN 3-89090-539-0, Preis: 69 Mark

Systemprogrammierung in C

Neben der inzwischen in immer größerer Zahl erscheinenden deutschen Amiga-Literatur sind die Übersetzungen englischsprachiger Werke für Amiga-Anwender und -programmierer interessant. Zu dieser Sparte zählt auch »Systemprogrammierung in C« (Originaltitel: Inside the Amiga) von John Thomas Berry.

In der »Einführung« findet der Programmierer eine Kurzbeschreibung von Hardware und Betriebssystem des Amiga sowie einige Anmerkungen zum Aufbau des Buches. »Die Programmierung des Amiga« stellt Arbeitsweise und Anwendung eines »C«-Compilers (Lattice) innerhalb der Systemumgebung des Amiga und einige spezielle Datentypen, die bei der Programmierung zum Einsatz kommen, vor. »Wie man Intuition benutzt« — dieser Titel spricht für sich: Hier werden die wichtigsten Intuition-Elemente (Screens, Fenster, Menüs, Gadgets, Requester) und deren Programmierung vorgestellt. Dabei geht der Autor auch recht ausführlich auf die Programmierung von Ein-/Ausgabeoperationen (IDCMP, console.device) unter Intuition ein. Das Kapitel »Prozeßkontrolle und AmigaDOS« bietet einen Einblick in das Multitasking-Betriebssystem des Amiga und erklärt die wichtigsten Grundbegriffe seiner Programmierung. Für den schon erfahrenen Programmierer ist diese Erklärung sicher ausreichend; der Anfänger wird sich damit eher schwer tun. »Zeichnen mit Intuition« erklärt die Zeichenfunktionen die dem Programmierer unter Intuition zur Verfügung stehen. Zusätzlich werden noch einige der Grafikfunktionen der Grafikbibliothek des Amiga vorgestellt. »Sprites bewegen« behandelt alle nötigen Funktionen und Strukturen, um Sprites zu definieren und in eigenen Programmen einzusetzen. Mit »Die Programmierung von Klängen« demonstriert der Autor die Verwendung des Amiga zur Klangerzeugung. Dabei werden erst einige allgemeine Grundlagen zur Erzeugung von Klängen und danach die Amiga-Programmierung im speziellen vorgestellt. Der Abschnitt »Künstliche Sprache« widmet sich der Sprachausgabe mit den im Amiga zur Verfügung stehenden Möglichkeiten. »Programmierung mit Disketten-Dateien« bildet den Abschluß des Buches.



Das Buch ist trotz der in einigen wenigen Kapiteln zu kurz geratenen Funktionsbeschreibungen jedem C-Programmierer zu empfehlen, denn er wird die in diesem Werk zusammengefaßten und damit schnell verfügbaren Informationen bei der Programmierung schätzen.

(Ingolf Krüger/pa)

John Thomas Barry, Systemprogrammierung in C, Tewi Verlag GmbH, 454 Seiten, ISBN 3-921803-71-3, Preis: 59 Mark

AMIGA
COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »Amiga« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 4 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **Februar-Ausgabe** (erscheint am 27. Januar 88): Schicken Sie Ihren Anzeigentext bis zum 17. Dezember 87 (Eingangsdatum beim Verlag) an »Amiga«. Später eingehende Aufträge werden in der **März-Ausgabe** (erscheint am 24. Februar 88) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 4 Zeilen mit je 40 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, Amiga« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 12,— je Zeile Text veröffentlicht.

Private Kleinanzeigen

Suche: Software

Suche günstige Software für Amiga, Spiele und nützliche, sinnvolle Progr., Listen bitte an M. Kirbach, Primelstr. 2 a, 8000 München 21

Suche zuverl. Tauschpartner für Amiga-Softw. (suche günstiges Hardwarereference-Manual) ★★★Tel. 0231/310464 (Dirk)★★★ ab 15 Uhr

Suche Software für Amiga 500. Listen an Harald Kadagies, Schildstr. 8, 2800 Bremen 1/ suche außerdem Tips + Information über deutsche Sprachsynthese (Programm?)

Amiga-Einsteiger sucht Software aller Art (besonders Spiele). Schreibt an: Jörg Schäfer, Igelweg 16, 6090 Rüsselsheim

Suche Software für Amiga 500 (Spiele und andere Programme). Listings bitte an M. Blinzinger bei Scheffl, Karlstraße 16, 8122 Penzberg

Suche Tauschpartner auf Amiga 500! Schreibt an: Wolfgang Stadler, Reinbekerstr. 18 A, 2000 Stapelfeld

Amiga
Tausche und kaufe Amiga Software! Listen an Udo Volk, Erlenbacher Str. 45, 6120 Erbach, 100% Antwort

Suche zuverlässige Tauschpartner für Amiga Software! Schickt Eure Liste an Thomas Hohoff, Am Zippen 37, 4600 Dortmund 1, Tel. 0231/511320 (ab 18 Uhr) 100% Antwort

Suche gebrauchte Amiga Originalsoftware. Angebote bitte an Detlef Erdmann, Kasseler Str. 84, 3502 Vellmar. Suche außerdem Kontakte zu A2000 Benutzern

Suche PRG zur Rechnungserstellung mit dem Amiga 500 oder 2000. Eckhard Meißner, Klaskamp 3, 3000 Hannover 21

Suche Manx AZTEC C-Compiler Version 3.4a und eventuell auch andere Amiga-Software. Angebote an: Peter Weiland, Ernst-August-Str. 7, D-3109 Wietze

Amiga-Software Amiga-Software
Tausche Software und suche Anleitungen. Habe immer neueste Software. Tel. 040/218695

Suche billig Software für Amiga 500, Tel. 0441/69994 oder Holger Knoblich, Glockenstr. 6, 2903 Bad Zwischenahn

Suche Software zur differenzierten farblichen Auswertung von IFF-Grafiken, insbesondere prozentuale Berechnung der Farben.
Dr. med. L. Busch, Baar-Klinik, Alte Wolterdinger Str. 68, 7710 Donaueschingen

Suche neueste Version der Kickstart-Disk 1.2, B. Hoffmann, Paul-Gerhardt-Ring 27, 1000 Berlin 20

!!! Tauschpartner für Amiga gesucht !!!
Suche zuverlässigen Tauschpartner! Habe Top Games! Thorsten Wendlandt, Ginsterweg 28, 7218 Trossingen, Tel. 07425/8573

Suche jegliche Grafik-Software (Deluxe, PD, Animation, CAD, Demos) mit Anleitung. Info bitte an Frank Liening, Erich-Nörrenberg-Str. 6, 5860 Iserlohn, Tel. 02371/27996

★★ MILAN ★★ is searching for swappartners (A-500-Soft) in whole Europe. If you are interested, send lists or disks to:
Milan, P. O. Box 65203 C, D-5400 Koblenz 1

Software gesucht. An **allem** interessiert, da neu auf Amiga. Kaufe gebr. Originale, Public Domain, usw. Programmlisten und Angebote an PF 145, 7405 Dettenhausen

Suche Software für Amiga
02202/85596

Kaufe gebr. Originale. Suche u.a.: Return to Atlantis, Tass Times, Alien Fires, Bard's Tale, Uninvited, Brataccass, Faery Tale. Tel.: 05204/8196 (öfters versuchen)

Einsteiger sucht Software für Amiga 500. Spiele und Programme (besonders Videoscape-3D) gesucht. Listen an: L. Sebastian, Regensburgerstr. 4, 2800 Bremen. Amiga? Yeah!!

Suche Software für den schönsten, schnellsten, stärksten, buntesten, lautesten und dynamischsten Computer der Welt. Erst ab 17.00 Uhr. Tel.: 02171/49761 (Michael)

Student sucht Kontakt und Spiele für Amiga 500, Noutsa Francis CDC, Luisenplatz 2, 7760 Radolfzell

Suche Software für Amiga 500, Andreas Fröhlich, Leimstr. 37, 8940 Memmingen

Suche Amiga-Software + Anleitungen, R. Meis, Am Hofacker 18, 4300 Essen 18

■■■■■■■■■■■■■■■■■■■■■
Habe & Suche
Topsoftware Topsoftware
■■■ Call 05222/60453 Call ■■■

Ich suche noch Tauschp. aus Dortmund. Habe z.B. Firepower, Pinballwizard, Testdrive, Amigas, Impact, alle neuen Copies und Anwenderprg. Nur Samstags ab 10 Uhr: 0231/173229

★★★ ★ Amiga 500 ★ ★★★
Suche, kaufe, tausche Software aller Art.
Tel. 04551/2807 (Torsten)
★★★★ ★ Amiga 500 ★ ★★★

Aktienprogramm für A 2000 ges.
Eduard Braun, Am Grünland 39, 8900 Augsburg, Tel. 0821/708109

Suche! Prowrite, Analyse V2.0, Adventure Comstr. Set, Deluxe Print, Grabbit, Prism. (event. Anleitung) und weitere Torsoft, Tel. 040/8807113 ab 15 h

Alter Computerfreak und neuer Amiga 2000 Besitzer sucht Software aller Art. Ralf Sauter, Wetterkreuzstr. 103, 7950 Biberach, Telefon 07351/23201

■■■ Amiga-News-Amiga ■■■
Ständig aktuellste Amiga-Software
■■■ Tel. 07563/2273 ■■■

Suche zuverl. Tauschpartner für Software! Suche besonders Spiele und einen C-Compiler. Listen an Alex Regnery, In der Olk, 5559 Klüsserath oder 06507/4479 ab 15 h

Amiga-Programmierer! Wer programmiert mir kaufmännisches- und Horoskop-Programm gegen gute Bezahlung? Leitfaden vorhanden. An: Franzwa, Postfach 24, A-5015 Salzburg

Suche Amiga-Soft: Tausche oder kaufe; suche besonders: Sculpt 3D; Aztec-C; (Turbo-)Pascal; Liste an: Stefan Trampler, Bonner Str. 92, 5483 Bad Neuenahr, Tel. 02641/1452

Amiga 2000 Anfänger sucht Software (Public Dom., Spiele) Kontakte zu Clubs, R. Dörr, Postfach 100808, 6800 Mannheim 1

★★ Amiga 500 Neuling sucht Tausch- ★★
Partner. Habe nicht viel Software.
Schickt Eure Listen an: Roland Schröder, 19, Quartier Wendel, 3881 Schifflingen, Lux.

Ausland

Suche/tausche neue Software. Auch PD; Weekend (ab Fr. 19 h) call Darth Vader, Tel. Austria = 0043/02231/2850. Bis bald!

Suche Tauschpartner für neueste Amigasoftware. Auch Ausland-Kontakte erwünscht! Listen und Disks an: André Grand, Sternmattstr. 117, CH-6005 Luzern/Schweiz

Jack/CH ★★★ Amiga-Stuff ★★★ Jack/CH
We search Swapping-Partners or best contacts in Germany!!! Write to: Jack Colton Club, Postfach 23, 5312 Döttingen/Suisse

Suche Tauschpartner für Amiga 1000 Programme. Listen bitte an P. J. Booy, Napo 86IF9, 3509VP Utrecht, Holland. Bitte 80 Pf. Marke für Antwort und Liste.

★★ Amiga Schweiz Amiga ★★
Wir suchen Tauschpartner. Brief an: Manfred Seiler, Auweg 21, CH-4450 Sissach oder Werner Zulauf, Bützenenweg 88, CH-4450 Sissach

Einsteiger sucht Amiga-Software, zahle pro Diskette Software 5-10 DM. Schickt Eure Liste an: Marcel Schils, Schatteheiweg 39, 6343AG Klimmen (NL)

Suche Tauschpartner in aller Welt. Habe immer das neueste, Ammon Andreas, Ach Siedlungstr. 43, A-6900 Bregenz

Suche alles an Software (auch Public Domain) und Anleitungen für meinen Amiga. Melden bei: Böckle Gerhard, Hauptstr. 25 a, A-6805 Gisingen, Tel. 05522/265973

■ Amiga-CH-Amiga ■
Suche Tauschpartner! Neueste Soft vorhanden! Thomas Rietsch, Postfach 133, CH-5415 Nussbaumen oder Tel. 056/824893!

Suche Tauschpartner für Amiga. Liste an: J. S. D., 78 rue Belair, L-3820 Schifflange, Luxemburg! (Auch Anfänger)
★★★★ J. S. D. For ever! ★★★★

Hi, suche **Amiga-Userinnen** in der ganzen Welt für Software und Erfahrungsaustausch. Dave Wäger, Niesenblickweg, 3655 Sigriswil (CH) ■ I'm waiting for answer!

Biete an: Software

Verkaufe original Shanghai,Starglider und das Amiga Basic Buch von Markt & Technik + Diskette. Tel. 05102/2424

Verkaufe **ORIGINAL** Uninvited u. Faery Tale Adv. zu jeweils DM 40,— für den Amiga 500. T. 089/5701454 tägl. ab 19 Uhr (fragt n. MARTIN)

Amigasoft immer neu!!!!
The Giant 040/214271

PUBLIC DOMAIN ★★★
Biete Fish-Disks Nr. 51-80
Tel. 02102/35384 (ab 20 Uhr)

■ The new Dimension!!! Amiga-Software ■
We are searching for new contacts on Amiga all over the world. Write to: TND/Postf. 2363/D-4788 Warstein-Belecke

Deutscher Free-Soft-Club-Amiga ★ Info gegen 0,80 DM Porto ★ von: FSC c/o Dieter Will, Postfach 2824, 2350 Neumünster

Ich suche einen...., na ja gut zwei...., ok von mir aus können es auch mehr sein, Tauschpartner für Amiga-Software. Bitte melden unter: 05103/7507 (Driiiiingeeend)

Verkaufe nur Originale (!): Arazok's Tomb + Lös.; B. Cauldron + Lös., Guild of Thieves + Lös.; Knight ORC; Land of L. Lizards. Keine Einzeltips (!!) 02134/32855

Verkaufe 12 Disketten voller Programme, Bilder und Demos (alles Public Domain) für nur 70 DM (bei Vorkasse inkl. Porto)
P. Weiland, Ernst-August-Str. 7, 3109 Wietze

★★★ Topaktuelle Amigasoftware ★★★
Suche Tauschpartner, habe nur das Neueste. Eventuell auch Verkauf von Originalen.
Tel. 02369/22459

Orig. Prog. inkl. Handbuch: Aegis Draw + DM 250, Lattice C 3.1 DM 200, MCC Pascal DM 150, Uhr für A 1000 DM 120. Tel. 030/6024249

■■■ Amiga-Software ■■■
Write to Rainer Dietmer, Am Sonnigen Hang 2, 6500 Mainz 32
■■■■■■■■■■■■■■■■■■■■■

Suche Tauschpartner
Tel. 04531/84112

Powerstation — Software — Production
Call us directly:
0711/2626692 — ask for Jörg

Suche und biete Amiga-Software
02655/1031 Timo (14-22 h)

Verkaufe orig. Software zum fairen Preis: Guild of Thieves, Defender of the Crown, Champ. Golf, Worldgames. ■ Suche günstige 1-2 MB Erweiter. — Tel. 0211/792566

■ Tausche neueste Amiga-Software ■
und Anleitungen! Suche alles auf MS-DOS! Tel. (Germany) 05561/6915 New Man« ★★★

Amiga-Anfänger sucht/kauft orig.! Software (insb. Spiele) zu fairem Preis. Angebote an P. STAHNKE, Großenbuschstr. 91, 5205 St. Augustin-2, od. Tel. 02241/332191

Steve is back on A2000/früher C64/128. Verschenken kann ich wieder was!! Alle Software! Nur Brief — Stefan Thomas, Friedensring 44, 5014 Kerpen/Greets to Megaf.!

★★★★★★★★★★★★★★★★★★★★
Tausche/verkaufe Amiga-Soft
Tel. 02181/499361 Mo-Fr 18.00-20.00
★★★★★★★★★★★★★★★★★★★★

Tausche neueste Software! Suche auch billige Hardware! 04101/43254 Oliver

Achtung:

Wir machen unsere Inserenten darauf aufmerksam, daß das **Angebot**, der **Verkauf** oder die **Verbreitung** von urheberrechtlich geschützter Software nur für Originalprogramme erlaubt ist.

Das Herstellen, Anbieten, Verkaufen und Verbreiten von **»Raubkopien«** verstößt gegen das Urheberrechtsgesetz und kann straf- und zivilrechtlich verfolgt werden. Bei Verstößen muß mit Anwalts- und Gerichtskosten von über DM 1 000,— gerechnet werden.

Originalprogramme sind am Copyright-Hinweis und am Originalaufkleber des Datenträgers (Diskette oder Kassette) zu erkennen und normalerweise originalverpackt. Mit dem Kauf von Raubkopien erwirbt der Käufer auch kein Nutzungsrecht und geht das Risiko einer jederzeitigen Beschlagnahmung ein.

Wir bitten unsere Leser in deren eigenem Interesse, Raubkopien von Original-Software weder anzubieten, zu verkaufen noch zu verbreiten. Erziehungsberechtigte haften für ihre Kinder.

Der Verlag wird in Zukunft keine Anzeigen mehr veröffentlichen, die darauf schließen lassen, daß Raubkopien angeboten werden.

Private Kleinanzeigen

Verkaufe supergünstig Amiga-Software! Suche auch Tauschpartner(innen)! Habe immer aktuelle Software. Schreibt an: L. Stevenson, Stockholmstr. 23, 2300 Kiel

---

Tausche + kaufe Amiga-Software!
Listen an: Arno Keiler
Felix-Schefflerstr. 24
8226 Altenmarkt

Big-Byte sucht Tauschpartner!! Allerneueste Software vorhanden! Tel. 05250/283 Ralph Dinslage; Kirchplatz 6, 4795 Delbrück. Verkaufe auch Originale!!!

★★★ Amiga Grafik u. DTP Freaks ★★★
Umfangreiche Grafik (1000 Bilder) u. Zeichensatz (150)-Sammlung. Info gegen Rückporto. R. Hobbold, Gleisstr. 14, 43 Essen 11

Search **Amiga 1000-Software + Contacts** to other Amiga-Freaks! Call 0231/465986 or write to: Frank Kwiatkowski, Rispenstr. 27, 4600 Dortmund 30! **TRONICA** thanks you

---

Suche zuverlässigen Tauschpartner für Software (bes. Spiele). Suche bes. C-Compiler. Listen an Alex Regnery, In der Olk, 5559 Klüsserath oder wählt 06507/4479

Verkaufe: MS-DOS Emulator sowie MS-DOS Systemdisk mit GW-Basic und MS-DOS Handbuch für 110,— DM, Tel. 07621/48923

Tausche orig. Softw., Tel. 04122/55741, DPaint, DVideo, Textcraft, Instant, Music, M. Madness, Temp. o. Asp., Am. Karate, Flight II gegen Aztec-C-Comp., Comm od. Developer, Tel. 04122/55741

★★★★★★★★★★★★★★★★★★★★★
Suche Tauschpartner für aktuelle Amiga-Software!! Tel. 02403/21203 ab 19 Uhr
★★★★★★★★★★★★★★★★★★★★★

★★★★★★★★★★★★★★★★★★★★★
Habe neueste Topsoftware für Amiga
Call: 0251/217773
★★★★★★★★★★★★★★★★★★★★★

Ausland

• Anfänger — Profis • Software aller Art super-preisgünstig abzugeben (Tausch?!) Erfahrungsaustausch!!! Antworte zu 100%, M. Sepp, Roseggerstr. 4, A-6020 Innsbruck

---

★★★ Österreich ★★★
Amiga-Software günstig abzugeben.
Tel. 0222/3928583

Amiga Freak want to swap software and ideas. Write to Morten Petersen, Stejlbjergvej 3,1, 6000 Kolding, Denmark, or call DK-05504437 after 18.00

The Clan Presents: Amiga-Soft in Schweiz
Neueste und beste Software zum Superpreis + auch Hardware. Roger Wattenhofer, Buelstr. 7, 8854 Siebnen/CH. Tel.: 055/642065!

Suche Tauschpartner für Amiga-Software. Ich habe mehr als 150 Disks. Schreibt an Frowin Lutz, Kugelangerg. 1, A-6060 Hall oder ruft einfach an: 0043/5223/2191

★★★★★★★★★★★★★★★★★★★★★
Do you want to swap newest Soft? OK!! Contact me: Erich Küng, Postfach 114, CH-3098 Köniz/Switzerland

---

★★★ Top Soft ★★★
C-64 and Amiga, call me 062/692535, Switzerland

■■■ Amiga Suisse Amiga ■■■
Hallo Freaks! Tausche Software!
Immer das Neueste! So call:
061/960420 Schweiz Arjan Uerlangen

Suche: Hardware

Suche für Amiga 500 zus. Floppy-NEC oder Commodore und Speichererweiterung 512 K. Suche zuverlässige Tauschpartner für Amiga Progr., Tel. 0221/7089049 nach 18 Uhr

---

■■ KAUFE ■■ KAUFE ■■ KAUFE ■■
Amiga 500, 1000 o. 2000, Speichererw., Monitor, Drucker, Laufw., Zub., Prog., ...
Tel. 05631/8360 oder 06043/2450, Dieter...

■■■ KAUFE ■■ AMIGA ■■■
Suche Amiga 500 oder 2000 + Monitor 1081 + Drucker
■■■ Tel. 05621/71019 abends ■■■

★★★★★★★★★★★★★★★★★★★★★
Kaufe defekte Amigas zu Höchstpreisen. Suche außerdem defekte C64, Floppies usw.; suche PC Board f. A2000, Tel. 04351/43670
★★★★★★★★★★★★★★★★★★★★★

A 500 — Suche günstig: Floppy, Disks, Speichererw. mit Uhr, Sounddigitizer. Alt o. neu nur 1A Zustand, call me:
★★★ 0521/335729 Adrian (17-20 Uhr) ★★★

Der Hammer Amiga KickstartV1.1 Emulator für Amiga 2000 u. 500 DM 18,—. Suche immer gute Amiga Software! Th. Wirz, Höhenweg 98, 5300 Bonn 1

Amiga-Einsteiger sucht Druckertreiber oder EPROM zur Anpassung von CP-80X an Amiga 500. Preis nach VB. Melden bei 030/6018392!

Amiga Einsteiger sucht lauffähige Software für A2000. Club Raum Dortmund gesucht. Listen an Dieter Thiele, Köslinerstr. 5, 4620 Castrop-Rauxel 4, Tel. 02305/75938

Suche den neuen MCC-Macro-Assembler und den Lattice-C-Compiler (beides mit Anleitung). Angebote an: V. Schenk, Noblingweg 1, 7840 Müllheim 16, Tel. 07631/13143

Ausland

Suche
64-Emulator, Macintosh-Emulator, ST-Emulator für AMIGA 500 — Thomas Mouzakitis, Tsamadou 4, 10683 Athina — Greece

---

Suche C-64-Emulator u. Macintosh-Emulator, Midisoftware und Hardwarebauanleitungen. Tausche Software: (ST-Emulator + IBM-Emulator), S. Himmelbauer, A-3741 Leodagger 27, Österreich

---

★★★ Hilfe ★★★ Wer schenkt einem armen CH CH Computerfreak einen Computer. Bin fasziniert von Amiga + Atari. Habe kein Geld, ehrlich! M. Lüthi, Ringstr. 3, 9302 Kronbühl

Biete an: Hardware

AMIGA 1000-768 K-RAM-Erweiterung DM 350,—; **GEMINI 10X**-Centronics-Drucker DM 400,—, C64-Software/Literatur ab DM 15,—; bei: E. Hecht, POB: 1273, 8482 Neustadt; 09602/2515

Amiga 1000 mit Softw. 799 DM, 1 MByte RAM Erw. für 500er, VB 230 DM, Tel. 0271/332423 ab 18 Uhr

---

Ideal für Einsteiger! Verkaufe für AMIGA 500 TV-Modulator 520 (ungebraucht) + Top-Programme. Bono Lapsien, Dorstenerstr. 110, 4690 Herne 2 (supergünstig) a. einzeln

---

Verkaufe Epson Drucker RX 80, Drucker ist neuwertig VB 280 DM, Tel. 02526/2883

★ Amiga 1000 ★ Neu ★ Amiga 1000 ★
Speichererweiterung Amiga 1000 auf 1 MByte für 250,—, Tel. 040/6434213
★★★ Einfach mal anrufen! ★★★

Amiga 2000 m. PC/XT, 6 Mon., 1 MB/512 K, 2 LW 3,5" 1 LW 5,25", RGB-Monitor 1081, 5 Handb., 60 Disk. VB DM 4300,—, R. Wirth, Pappelweg 17, 2300 Kiel, 0431/681759

Verkaufe AMIGA 1000; 1 Megabyte
mit Monitor und 50 Disk Software und Diskbox. Auch einzeln! Preis nach Vereinbarung, Tel. 07031/273416 nach 16.00 h

---

Speichererweiter. für Amiga-500 auf 1 MB inkl. Uhr u. Batterie einsteckfertig für untere Klappe VB 295 DM ★★ Aztec-C V3.4 a Commercial m. Buch u. dt. Tast. 0201/358239

Amiga 2000 (1 Monat) inkl. 2 3½" Laufw. sowie einem 5¼" Laufw., Monitor 1081, Boot-Selektor, abschaltb. Fast-RAM, Softw., Lit., etc. DM 3500,— ab 31.10.87, Tel. 05657/1650

Verk. ★ **Amiga 1000** ★, 512 KB + Handbuch + dazugeh. Software + Druckerkabel + orig. Anwenderpaket (Craficraft, Textcraft, Amiga Buch) ★ 1790,— DM ★ T. 0681/852209 ab 15 h

---

A1000 + 1081 1800,—, Star NG 10 480,—, DF1: 300,—. Interesse? (02105) 71282 oder W. Scharfe, Marienburger. 23, 4005 Meerbusch 1
★ Alles einwandfrei ★

Verkaufe A1000, 1 Megabyte, Preis: VHS
Suche A2000, evtl. m. Zubehör, ohne Monitor. Preis <1800 DM.
Andreas Nachreiner, ☏089/3234176

NEC-Zweitlaufwerk im Gehäuse, anschlußfertig inkl. 20 Disks (voll) DM 299,—
Amiga Reference Manuals, 4 Bände, neu, DM 200,—, Tel. 02871/37375 ab 17 Uhr

---

Verkaufe: C128 + Abdeckhaube, C1541 Floppy + Abdeckh., 50 Disketten BASF, 32 x Input 64 + Kassetteninterface, komplett o. einzeln, Tel. 02542/4344 o. 5312

Verkaufe: AMIGA 1000 mit 512 KB + Farbmon. + 2. Floppy + Sounddigitizer + Software + Disks + Bücher + Diskettenboxen + Druckerkabel + Anl. gegen Höchstangebot, Telefon: 05130/5161

Verkaufe: Speichererweiterung + Uhr für Amiga 500. Preis: 220 DM. Amiga 1081 Farbmonitor für 700 DM, Tel. 0251/211744

Amiga 1000 User
Speichererweiterung auf 512 KB für 100 DM abzugeben. 02203/23009

Verk. Commodore-PC 10 mit MByte Harddisk 640 K-RAM, RS232, Parallelport, Farbkarte u. reichlich Software (alle Orig.), Farbmon. Taxan ½ Jahr VHB 3500 DM, 05423/41799

Verk. Epson komp. Drucker mit Schönschr. 100 CPS installieren in Preferences als Epson mit Kabel für Amiga und Papier etc.
A. Ropers, Tel. 06571/8440 VB 350,—

Diverse Hardware für den Amiga:
Z.B. Super Digitizer, ECB-bus, IO-Karten usw. als Bausatz o. Fertigg. Info gegen Porto bei M. Wechsung, Goethestr. 8 a, 33 Braunschweig

---

CBM 610 komplette Anlage (DM 2200) (CBM 8250 LP, MPP 1361, Phil. Mon. mit MACRO-Basic) zu verkaufen! Rheindorf, Stettiner Str. 33, 7922 Herbrechtingen

---

Verkaufe »Amiga 500« neu mit Garantie, wegen Systemwechsel DM 1098,— VHB. Anfragen bitte an Michael Schleith, Tel. 07621/63396

Sidecar m. Tandon-HC 20 MB, 6 Mon. alt für DM 2000,— mit AGA-Karte u. Monocr. Monitor (bernst.) DM 2700,— (emul. alle Spiele), Sidecar allein DM 1200,—, Tel. 0228/464307

★★★ !!!!! ★★★ Verkaufe Amiga 2000 für 499 DM! Tel. 02171/53767!!! Tag und Nacht!

Echtzeituhr extr. genau, steckb. an Port 2 (durchgef.) NP 129 jetzt 60 DM, Program. Handb. 1 + 2 v. Mortimore NP 99/Stck. nur 65 tausche PD + Util., Tel. 02261/28141 Ralf

---

Zu verkaufen!
Amiga, Floppy, Monitor, Drucker, C-64, 1541, Monitor, Drucker, Zubehör, Bücher. Anruf lohnt! 07125/8317 ab 17.15

Grafiktablett Easyl für Amiga 1000:
druckerempfindlich, mit Software, neuwertig für nur 600,— DM zu verkaufen!!!
Rainer Wolf, Tel. 02541/2874 ★★★

■■■ **Amiga 2000 — RAM-Ausbau** ■■■
intern von 1 MByte auf 1,5 MByte
mit Einbauanl. inkl. Versk. DM 185,— (aus Sammelbest.) Tel. 05481/81451 ab 19 Uhr

Ausland

Verkaufe Fujitsu Matrix-Drucker DX2200 + Anleitung (deutsch) für 1200 DM!
Georg Eberle, Im Grund 22, CH-6462 Seedorf (Schweiz), Tel. 0041/44/25093

Verschiedenes

Suche Handbücher für Logistix, Aegis Draw Plus, Dynamic Cad; habe Superbase
Georg Halfas, Hermeskeiler Str. 9, 5000 Köln 41, Tel. 0221/434527

■■■■■ Amiga JUICE ■■■■■
DAS Amiga Magazin auf Diskette!! Amiga-JUICE ist Public Domain und jeder macht mit!! Für Infos: 05103/2146 ab 16 Uhr!

Suche Amiga Ausgaben 6/7 und 8/9
Ralph Peitz, Tel. 05323/78139

Verkaufe SYBEX-Buch: Amiga Programmer's Handbook DM 45; tewi-Buch: Amiga Systemprogrammierung in C (neu; noch unbenutzt) DM 40, Aztec, C-Anl. DM 30 VLB; Rainer: 0661/54772

Suche Kontakt, zwecks Infoaustausch, zu Amiga 2000 Usern oder Clubs, bevorzugt Raum Koblenz, Tel. 02601/1810

Suche deutsche Anleitung für: Videoscape, Aegis Drawst, Prowrite, Pagesetter, Superbase; Sidecar, Genlock, Realdigitizer ges., Anwendg. + Tausch, Tel. 04292/3126 (18.00)

Suche Kontakte zu netten AMIGA-Anwendern, auch Neulingen, zwecks Erfahrungs- und Programmtauschs. Ich nutze mein Gerät privat und geschäftlich. Michael Lange, Boppstr. 4, 1000 Berlin 61 — Tel. 030/6931028

★★★ Ricercar eV. ★★★
Der führende Verein für alle AMIGA, ST und C 128 User. Wir bieten mehr! Zeitschrift, PD-Software, Beratung, etc... Info: Ricercar eV., Dorstener Str. 31, 435 Recklinghausen, Tel. 02361/15943

Raum Flensburg: Amigos gesucht für Erfahrungsaustausch über Hard- und Software (insbes. Text + Pictures). Ab 17 h, Tel. 0461/65445

Wer verkauft mir Amiga Ausgabe 6/7 für DM 20. Suche Prüfprogramm, mit dem die komplette Hardware des Amigas geprüft wird. Tel. 06421/14871 ab 20.00 Uhr

---

An alle **AMIGA**-Besitzer im Raum Düsseldorf Süd. Suche oder gründe **Club**, Telefon: (0211) 712947

■■■ DRINGEND ■■■
SUCHE AMIGA-Magazin, Ausgabe 6/7 und 8/9, Tel. 0911/502568

Suche Kontakt zu Amiga-500-Besitzer im Raum Ravensburg. Werner Diesch, Vogelhäusle 1, 7980 Ravensburg. Verkaufe f. C64 1 Mega-Epromkarte von Rex-Datentechnik.

Kontakte zum Austausch von Erfahrung und Software im Raum Stgt. gesucht. Auch BRD und Ausl. Martin Gerlach, 7032 Sindelfingen, Leonberger Str. 29, Tel. 07031/801864

Suche Amiga-Benutzer zwecks gemeinsamen Einkaufs von Software (Compiler, Utilities etc.). Wolfgang Duwensee, Tel.: 089/683175

Ausland

Megaforce
Sander Hans, Traunaustr. 4, 4600 Wels in Austria, Tel. 07242/820064 ab 19 Uhr!!!!
Megaforce

Anno 1992 wird Großeuropa gegründet... in Straßburg gibt es 1 Commodore-Club wo dieses Europa mitbauen möchte. Schreiben Sie an Dietrich L., BP 7, F-67200 Oberhausbergen

Gewerbliche Kleinanzeigen

PD: 3 DISK m. SPIELE o. Anwend. o. n. WAHL 20 DM inkl. Porto; SCHEIN an: Paul Hackl, C-Fachh., Bauweberstr. 13, 8 Münch. 71, (089) 7559678

AMIGA-Broker: Börsenspiel für 4 Spieler, Maussteuerung, in deutsch und englisch, Anleitungen auf Diskette, min. 512 KB für nur 30,— DM (Schein oder Scheck)
A. Page, Im Löhken 100, 4355 Waltrop

★★★ **AMIGA BILDSCHIRMFOTOS** ★★★
Lassen Sie Ihre IFF-GRAFIKEN nicht im Archiv verstauben!!! Wir fertigen BILDSCHIRMFOTOS von Ihren Grafiken in ERSTKLASSIGER QUALITÄT — IN ALLEN GRÖSSEN. Farbecht, Superscharf, Brillant. Ideal als INDIVIDUELLES WEIHNACHTSGESCHENK. Prospekt von OPTIVISION, Heckenerstr. 16 ★ 5469 Windhagen

---

1,30 — 2 DM (o. Disk); 3,80 — 5 DM (m. Disk) PD-Software! Info-Demo-Disk 5 DM; M. Hain, Lehen 55, 8098 Pfaffing, 08039/3031 (ab 16 h)

★ AUGENSCHMAUS von A.U.G.E. 4000 ★
Alle Tips, Tricks u. Prgr. vorher getestet. Auch US-PD. Disk ab 4,—. Infodisk 5,— Schein, AUGE 4000, Moltkestr. 115, 4 D'dorf 30

AMIGA-Finanzbuchhaltungssoftware
Testdiskette/Softwareprobe für 15,— bei MICROTEC, Rigaweg 1, 3300 Braunschweig, Tel. 05309/1466. Weitere Händleranfr. erwünscht!

★★★★ **AMIGA, C 128, C 64** ★★★★
!! Ständig Sonderangebote !! — auch für unser Angebot an Peripherie und Zubehör! Ebenso in unserem Programm: Hardwareerweiterungen + EPROM-Brennerei! Katalog gegen 1,40 DM Porto bei: Ewert & Huwer, Hard & Software Vertrieb, Birkenweg 21, 7637 Ettenheim 1, Tel. 07822/4696

Fish-Disks Nr. 1-30 nur 120 DM inkl. Disks, Porto u. Verpackung! V-Scheck an: Rudolf Sondermaier, Parkstr. 38, 8025 Unterhaching

---

Disketten NoName 3,5"1DD-DM 27,50
Disketten NoName 3,5"2DD-DM 31,59
AMIGAS-DRUCKER-ZUBEH. ELEKTR. BAUTEILE. Preisliste anfordern: W. Niemann Walsumermarkstr. 140, 42 Oberhausen 11

AIT-Amiga Idealisten Team-AIT
Über **400** Public Domain Disks, Top aktuell: Über **98** Fish, Faug, Taifun...
Auge, Amicus, Panorama, Casa, Icpug...
Super PD-Info: 2 Disketten = 12 DM = Beschreibungen >1200 KB, >500 Screens
Super Preis: 5 DM pro Disk ohne Staffel; ab 10 Stck. inkl. Porto etc. Wir tauschen auch: zwei zu eins. AIT-M. Rönn, Ziegeleiweg 32, 3257 Springe 4 (05041-8229)

DIGI-SERVICE, Digitalisierung Ihrer Bilder, IFF- u. PAL-Format, Farbe (auch H.A.M.) u. s/w, starke Vergrös. mögl., Infos anf. bei K. Juris, Bahnhofstr. 106, 6392 Neu-Anspach

---

AMIGA-Laufwerke 3½" DM 320,— 5¼" DM 365.— Tel. 07121/61882

PUBLIC DOMAIN für Amiga & IBM
Tiefstpreise + 24 Std. Versand
Katalogdisk gegen 5,— Vorkasse
Funkcenter Mitte GmbH
Klosterstr. 130, 4 Düsseldorf 1
Tel. 0211/362522
Mailbox 0211/360104 — 18-9 Uhr

DAS AMIGA-GLOSSAR TEIL 4

Der vierte Teil unseres Glossars bringt unter anderem interessante Informationen für diejenigen Programmierer, die sich näher mit dem Betriebssystem des Amiga beschäftigen, sich für das Dateiformat IFF interessieren oder Informationen zur Icon-Verwaltung des Amiga benötigen. Die mit dem Pfeil (->) gekennzeichneten Begriffe finden Sie als eigenes Stichwort innerhalb des Glossars.

(Davin Twigg-Flesner/pa)

Die einzelnen Teile des Glossars zum Heraustrennen bilden ein umfassendes Nachschlagwerk der gängisten Begriffe rund um den Amiga. Sammeln Sie mit.

hue Farbton. Die Farbvielfalt des Amiga mit 4096 Farben zu beschreiben ist insofern unkorrekt, als es sich dabei um 4096 Farbtöne handelt, die aus den Grundfarben Rot, Grün und Blau gebildet werden.

Eine besondere Variation des Themas »hue« ist das sogenannte »artifacting«. Hierbei entstehen als Folge der Interaktion benachbarter Farbtöne neue, in den Farbregistern nicht enthaltene Farbtöne. Aufsteiger vom VC 20 kennen vielleicht diesen Effekt vom Superexpander her: bei der zufälligen Farbauswahl für die COLOR-Parameter entstehen gelegentlich verschiedene Grautöne, die vom System aus eigentlich nicht vorhanden sein können. Diese Grautöne sind eine Interferenzerscheinung am Bildschirm, wie der begleitende Störton im Lautsprecher beweist.

hunk () (Teilstück, Segment) Ein Hunk ist ein Teil eines Objektcodemoduls. Je nach Inhalt unterscheidet man Hunks für Programmbefehle (Code), (initialisierte) Daten und nicht initialisierte Daten. Für besondere Zwecke eingesetzt werden Relokationshunks für die Adressenanpassung beim -> scatter load, symbol_hunks mit Listen von externen Symbolen, debug_hunks mit Symboltabellen und sonstigen für die Fehlersuche erforderlichen Daten. Am Anfang eines -> load file befindet sich ein hunkheader mit Daten für die Laderoutine (DOS-Loader); gegebenenfalls kommt noch ein hunk_overlay mit der -> overlay-Tabelle hinzu. Näheres über diese Bestandteile einer binären Datei ist dem Amiga-DOS-Manual (Technical Reference) zu entnehmen.

icon (Sinnbild, Piktogramm) Das grafische Symbol für ein Verzeichnis oder eine Datei auf der Workbench. Man unterscheidet fünf verschiedene Icontypen: Diskette, Schublade, -> Tool, -> Project und Trashcan (der Mülleimer). Das Diskettenicon und der Mülleimer symbolisieren besondere Verzeichnisse: das root directory und die Ablage für überflüssige Dateien. Im Gegensatz zur Menüoption »delete« erlaubt die Verwaltung des Mülleimers, »weggeworfene« Dateien wieder aus dem Eimer zu holen, um sie weiter zu verwenden. Das funktioniert jedoch nur solange, wie dieser Eimer nicht mit »empty trash« entleert worden ist. Das Diskettenicon und der Mülleimer können von der Workbench aus nicht gelöscht werden.

Die anderen Icons symbolisieren normale Verzeichnisse (user directory), Programme oder Programmdateien. Dem Diskettenicon, Trashcan oder Schubladenicon ist immer ein Window zugewiesen, in dem der Inhalt dieser Verzeichnisse — falls mit Icon versehen — sichtbar wird.

IconEd Das Icon-Editierprogramm der Workbench. Mit diesem Tool kann man eigene Icons (Bildsymbole) definieren, indem ein bereits bestehendes Icon in einen der Editierrahmen geladen und dann bearbeitet wird. Der zu ladende Icontyp muß dem Programm mitgeteilt werden. Dies geschieht entweder mit dem Namen oder, bei einem Diskettenicon, mit der Bezeichnung DISK. Wie beim Laden eines Icons von einem externen Laufwerk muß gegebenenfalls der vollständige -> path-Name angegeben werden. Das neu erzeugte Icon übernimmt die Systemdaten des alten Icons und ist somit auch vom selben Typ (Project, Tool..).

Neben dem IconEd gibt es weitere Editoren. Erwähnenswert sind: Zap-Icon und SetAlternate (auch als Icon-Merge zu finden). Zap-Icon besitzt eine Funktion, mit der sich »brushes« aus DeLuxe Paint als Icons speichern lassen. SetAlternate kann Icons mit unterschiedlichem Aussehen für die Darstellung vor und nach einem Mausklick (select status) erzeugen.

Hierzu werden zwei getrennte Icons miteinander verkettet. Jeweils eines für den normalen und den -> highlight-Zustand. Ein weiteres Hilfsprogramm für diese Zwecke heißt IconExec. Mit diesem läßt sich ein Programm ohne Icon (zum Beispiel ein CLI-Befehl oder ein -> batch file) über ein Icon aufrufen. Dieses Programm befindet sich zusammen mit SetAlternate auf der -> Fish Disk #12.

iconic interface Eine mit Icons betriebene Benutzeroberfläche (wie Workbench).

icon.library Die Sammlung der Systemroutinen, mit denen die Icons erzeugt, dargestellt und verwaltet werden.

IDCMP (Intuition Direct Communication Message Port) Ein Subsystem, das einem -> Window zugeordnet wird, um die Kommunikation zwischen zu diesem Window gehörenden Programmteilen und anderen Routinen oder -> devices zu verwalten. Ein IDCMP besteht aus zwei -> message ports, jeweils eins beim Programm und eins bei Intuition. Das Programm bekommt auf diesem Wege -> input events wie Texteingaben, Mausbewegungen, Fensterveränderungen und sonstige Aktionen mitgeteilt. Da die Daten vom input.device und nicht vom console.device kommen, sind es »rohe« Werte in den systemeigenen Codesequenzen, die vom Programm noch aufbereitet werden müssen. Informationsträger bei dieser Kommunikation ist eine Datenstruktur namens -> IntuiMessage.

IFF (Interchange File Format) Ein in erster Linie von -> EA entwickeltes Dateiformat für den Amiga, welches die Übertragbarkeit (portability) von Dateien zwischen unterschiedlichen Anwenderprogrammen gewährleistet. Grundlage des Formats sind sogenannte -> chunks. Solche Chunks bestehen aus einzelnen Datentypen (Text, Bilder...) und bilden aneinandergereiht die IFF-Datei. Eine ausführliche (englischsprachige) Dokumentation befindet sich auf der -> Fish Disk 16; Ergänzungsmaterial auf Fish Disk 44. Diese Informationen entsprechen der offiziellen Commodore-Dokumentation.

Gelegentlich taucht auch die falsche Bezeichnung »Interleaved File Format« auf. Dieser Name ist darauf zurückzuführen, daß der chunk-Typ »interleaved bitmap (ILBM)« als erster der breiten Öffentlichkeit bekannt gemacht wurde und irrtümlicherweise als Modell für alle IFF-Dateien angesehen wurde.

illustration data types Die Grafikroutinen von Intuition. Das sind Unterprogramme zur Gestaltung von Menüs, Gadgets und Requestern sowie Routinen zum Zeichnen von borders, images und text:

— Border (Umrahmung, Umrandung) ist eine allgemeine Zeichenstruktur für Polygone, mit der man einfache Linien und regelmäßige wie unregelmäßige Figuren darstellen kann.

— Image (Bild) ist die Struktur für die farbliche Darstellung eines Gegenstandes am Bildschirm; anders als bei border erfolgt die Darstellung nicht linien-, sondern pixelorientiert.

— Text (IntuiText) ist die Struktur für die Darstellung von Text. Anders als bei den meisten anderen Computer-

systemen wird Text beim Amiga als ein Element der Grafik behandelt.

Die obengenannten Strukturen sind Datenbereiche, welche die Parameter für das darzustellende Objekt beinhalten. Sie werden alle von den -> graphics primitives unterstützt. Die illustration data types können allerdings zu verketteten Listen zusammengestellt werden. Dadurch lassen sich mehrere Grafikelemente derselben Art mit einem einzigen Intuitionaufruf erzeugen.

image shadow -> masks

immediate mode der »Direktmodus« von Amiga-Basic und dem CLI. Jeder eingegebene Befehl (unter Amiga-Basic im Ausgabefenster) wird nach Betätigung der RETURN-Taste sofort ausgeführt (siehe interactive mode). Nach der Ausführung befindet sich der Amiga wieder im immediate mode und wartet auf einen weiteren Befehl.

Bei einigen Befehlen (insbesonders RUN) ist darauf zu achten, daß der Befehl auf alle Fälle sofort ausgeführt wird und nicht mit einem »?« für die Ausgabe eines -> command template von der Ausführung zurückgehalten wird.

include file (auch header file) Include- oder Headerfiles enthalten Daten (Konstanten, Strukturdefinitionen, Deklarationen externer Variablen, Makros...), die dem jeweiligen Sprachtyp gemäß definiert werden. Beispiel: Mit der Include-Datei <intuition.h> wird die Datenstruktur »window« definiert. Diese Struktur enthält unter anderem Daten über Länge, Breite und Art des Fensters. Bindet der Programmierer diese Datei mit #include <intuition.h> in sein Programm ein, so wird mit dem Laden und Ausführen dieser Datei die Struktur definiert. Der Programmierer erspart sich selbst damit die Definition und kann mit der Anweisung »MeinFenster STRUCT window;« eine eigene Variable von der Art einer Window-Struktur definieren. Das Beispiel zeigt, daß sich das Anlegen einer Include-Datei nur dann lohnt, wenn die jeweiligen Definitionen in mehreren Programmen verwendet werden können.

Bei der Angabe des Dateinamens in spitzen Klammern durchsucht der Compiler nicht das aktuelle Verzeichnis nach der Datei. Dies kann aber durch Einschluß des Namens in Anführungszeichen erzwungen werden.

info CLI-Befehl oder Workbench-Menüfunktion.

Als CLI-Befehl liefert INFO Informationen über die augenblickliche Diskettenausstattung des Systems. Angezeigt werden die verfügbaren Units (dfx:, jhx:, ram:) mit ihrem maximalen, belegten und freien Speicherplatz, die Anzahl der vom -> Diskvalidator festgestellten Lesefehler (soft errors), der Status (Schreibschutz aktiviert oder nicht) und Name der verfügbaren -> Volumes (Disketten). Zu den »verfügbaren« (mounted) Volumes zählen die SYS:-Diskette (-> system disk) sowie alle anderen Disketten, von denen Verzeichnisse (mit dem ASSIGN-Befehl) oder Daten (als Inhalt des Diskpuffers) noch im System eingebunden sind.

Anders als der CLI-Befehl gibt die Menüfunktion Info von den mit einem Icon versehenen Dateien Informationen aus. Bei einem Disk-Icon enthält das »Info-Blatt« bis auf Angabe der Lesefehler dieselben Daten wie die Ausgabe des CLI-Befehls. Bei Tools und Projects werden angezeigt der Dateityp (Project, Tool), die Größe der Datei in Bytes und Blocks, die Größe des notwendigen Stapelspeichers (Stack), der Status (Löschschutz aktiviert oder nicht), der Dateikommentar und Angaben über -> default Tool und Tool Type. Das Feld »Tool Types« wird oft verwendet, um dort allgemeine Informationen oder Übergabeparameter für die default Tools zu vermerken.

.info file (dot info) Die Icondatei einer Datendatei. Sie enthält die zur Darstellung des Icons und zum Starten des Programms von der Workbench erforderlichen Information für das Betriebssystem: Icontyp (einschließlich -> Kickstart-Diskette), Position des Icons am Bildschirm, das Fenster des Icons mit der Position untergeordneter Icons, den Stackumfang und andere. Um ein Icon »verschwinden« zu lassen, braucht man nur die Dateikennung ».info« zu ändern (Beispiel: .noinfo) oder die Infodatei zu löschen.

info.library Eine Sammlung von Systemroutinen, mit deren Hilfe die Icondateien verwaltet werden (auch als -> icon.library bekannt).

input.device (Eingabeschnittstelle von Intuition) Eine übergeordnete Systemschnittstelle, die Eingaben aus mehreren anderen .devices (gameport, trackdisk, keyboard) sowie von Peripheriegeräten zu einem einzigen »input stream« (Eingabedatenstrom) vereinigt und entsprechende »input events« signalisiert. Intuition überprüft den input stream zunächst auf relevante Daten und leitet ihn dann an andere datenerwartende Routinen weiter. Der Datenstrom wird nach Entnahme der relevanten Daten an die nächste Routine in der Verarbeitungskette über das -> IDCMP-Subsystem oder -> console.device weitergeleitet. Die erforderlichen Systemroutinen sind in den auch Texte verarbeitenden Libraries CList und CString zu finden.

interactive mode Der »Gesprächsmodus« des CLI. Ähnlich wie beim -> immediate mode werden Eingaben nach Betätigung der RETURN-Taste sofort ausgeführt. Allerdings kehrt der Amiga nicht in den Befehlsmodus zurück, sondern bleibt im Eingabemodus des aktiven Befehls, das heißt, der Befehl oder das Programm kann ohne neuen Ladevorgang weitere Eingaben verarbeiten. Einige CLI-Befehle verfügen über beide Ausführungsmodi.

interlace mode Verdoppelung der vertikalen Auflösung durch Zerlegen eines Bildes in zwei Teilbilder. Im normalen Betrieb zeichnet der Elektronenstrahl des Monitors ein Bild mit 200 Zeilen (256 Zeilen bei der PAL-Version). Diese Zeilenzahl ist nahe an der Grenze dessen, was der Elektronenstrahl relativ preisgünstiger Monitoren während des vertikalen Bildschirmlaufs darstellen kann. Im Interlacemodus besitzt ein Bild 400 Zeilen. Um diese auf den Bildschirm zu bekommen, zeichnet der Elektronenstrahl zunächst alle ungeraden Zeilen (1,3,5..), springt dann wieder nach oben und beginnt zwischen der ersten und dritten Zeile mit den geraden Zeilen (2,4..). Durch diese Technik werden pro Sekunde statt der 50 Zeilen (PAL-Version) im Normalmodus nur noch 25 Zeilen dargestellt. Das Auge registriert diese Minderung der« Geschwindigkeit als Flackern.

Ein mit Interlace erzeugtes Bild wirkt bei normaler Darstellung »zusammengestaucht«. Unter Version 1.2 kann dieser Modus über -> Preferences eingeschaltet werden. Der Modus wird aber erst nach einem -> reset aktiv. Damit ist Funktion »use« wirkungslos und »save« muß gewählt werden.

interpreter Ein Hilfsprogramm, welches die Zeilen eines in einer höheren Programmiersprache geschriebenen Programms nicht wie ein -> Compiler in Maschinencode übersetzt, sondern nach Erkennen eines Befehls in eine Programmroutine verzweigt, die diesen Befehl bearbeitet. Aufgrund der in jeder Zeile notwendigen »Interpretation« des Befehls, der Umwandlung der Variablennamen in Speicheradressen beziehungsweise ASCII-Konstanten in Fließkomma- oder Integerwerte ist ein mit einem Interpreter laufendes Programm sehr viel langsamer als compilierte Programme. Der Vorteil eines Interpreters ist der fehlende Aufwand für das Übersetzen und Linken des Quellprogramms. Das verführt jedoch nicht nur Programmieranfänger dazu, sich beim Programmieren weniger Gedanken zu machen, da man sich ja »schnell« mal ansehen kann, wie das Programm abläuft.

Intuition Das »user interface«, die Benutzeroberfläche des Amiga. Intuition ist die oberste Ebene des Betriebssystems. Für den Anwender durch die Verwendung von -> icons, -> Menüs, -> gadgets, -> requesters, -> screens und -> windows am einfachsten zu bedienen, weil keine Spezialkenntnisse über Programmiersprachen oder Computer erforderlich sind. Auch bei der Programmierung nehmen die Routinen der »intuition.library« dem Programmierer viel Ver-

waltungsarbeit ab. Deshalb sollten sich Einsteiger zunächst mit dieser Seite des Amiga befassen.

intuition.library Die Sammlung der Routinen, welche die Benutzeroberfläche Intuition verwalten. Wie aus dem entsprechenden -> .fd file ersichtlich, ist dies ein Bereich des Betriebssystems, der bei jeder neuen Version ständig verfeinert und erweitert worden ist.

IntuiMessage -> IDCMP

IntuiText -> illustration data types

IntuiTicks Die Zeitmeßeinheit von Intuition. Streng genommen werden 50 IntuiTicks pro Sekunde (laut Reference Manual) erzeugt. Unter Version 1.1 jedoch erfolgten diese doppelt so schnell. In der Version 1.2 ist dieser Mangel behoben.

i/o redirection Die »Umleitung« von Ein-/Ausgabeoperationen der CLI-Befehle. In der Regel benutzen die CLI-Befehle für die Eingabe die Tastatur und für die Ausgabe den Bildschirm. Durch die Angabe eines Dateinamens im Anschluß an die beiden Richtungsanzeiger »>«(Ziel, neues Ausgabegerät) und »<« (Quelle, neues Eingabegerät) ist das jeweilige »Gerät« nun die angegebene Datei. Das bedeutet, daß Eingaben nun aus einer Datei gelesen beziehungsweise Ausgaben in eine Datei geschrieben werden. Die Verwendung beider Richtungsanzeiger in einem Befehl bedeutet nicht, daß die Operation als Hintergrundtask ausgeführt wird. Dies ist nur mit dem Befehl RUN möglich (siehe pipelining). Die Gültigkeit der Umleitung beschränkt sich auf die jeweilige Befehlszeile. Die Richtungsoperatoren müssen stets vor allen erforderlichen Parametern der Befehlszeile — also direkt hinter dem Befehlswort — stehen.

ISO (International Standards Organisation) Das internationale Gegenstück des -> ANSI; 1946 gegründet. Die Vorschrift ISO-7 (beziehungsweise ISO 646-1973) entspricht nahezu der 7-Bit-Zeichendefinition des -> ASCII, hält aber bestimmte Codes für nationale Sonderzeichen frei. Diese Vereinbarung wird bei den verschiedenen nationalen Zeichensätzen der (Matrix-)Drucker angewendet. Die ASCII-konforme Codierung des Zeichensatzes wurde beim Amiga voll übernommen. Die erweiterten Steuersequenzen richten sich teilweise nach ISO und teilweise nach -> DEC.

Janus Der römische Gott der Türen und Torbögen; in der Regel als zweigesichtige (glattrasiert und bärtig) Gestalt dargestellt. Beim Amiga wird mit Janus eine Schaltung bezeichnet, welche die »Tür« zwischen dem Amiga (A2000) und einer hardwaremäßigen PC-Erweiterung (-> Sidecar, bridge board) verwaltet.

Die Janus-Schnittstelle besteht aus drei Sonderchips auf der PC-Karte und dem Amiga-Controller. Beide Systeme kommunizieren über einen als »dual port RAM« bezeichneten Bereich mit einem Umfang von 128 KByte. PC-Werte werden hier in drei verschiedenen Formaten (Byte, Wort, Langwort) in entsprechend vorgesehene Speicherbereiche abgelegt. Da die Adressen dieser Bereiche weit auseinander liegen, benötigt diese Schnittstelle insgesamt 2 MByte im oberen Bereich des -> fast RAM (siehe memory map). Ein auf der PC-Seite angeschlossenes Festplattenlaufwerk trägt als Kennung die Bezeichnung JH0: oder JH1: (statt DH0:/DH1:). Auf diesen Platten kann auch der Amiga eine »partition« (einen Teil) benutzen.

janus.library die Sammlung der Routinen für die Verwaltung der Janus-Schnittstelle.

jiffy Die von DeLuxe Video verwendete, 1/60 Sekunden andauernde Zeitmeßeinheit (siehe tick, IntuiTick).

jitter -> hires flicker

job Eine vom Computer zu erledigende Arbeit. Diese Bezeichnung gilt hauptsächlich für Rechenanlagen der großen und mittleren Datentechnik (mainframe) und beschreibt die Ausführung eines Programmes beziehungsweise einer Reihe von miteinander verknüpften Routinen, die — in der richtigen Reihenfolge abgearbeitet — eine sinnvolle datenverarbeitende Sequenz ergeben. Auf den Amiga bezogen wäre ein Job etwa die Erstellung eines Dokuments mit einem Textverarbeitungsprogramm, ein Compiliervorgang oder das Kopieren einer Datei oder Diskette.

kernel Der Kern beziehungsweise die unterste Ebene des Betriebssystems. Die Routinen, welche die Hardware-resources des Computers direkt ansprechen. Beim Amiga gehören hierzu alle -> WCS-residenten Routinen, die bis Version 1.1 von der Kickstart-Diskette eingelesen werden mußten beziehungsweise ab Version 1.2 im ROM (nur A500 und A2000) untergebracht sind. Der Kernelspeicher belegt zusammen mit dem -> boot ROM die obersten 256K der -> memory map.

kerning Der Abstand zwischen dem ersten Punkt einer Zeichenmatrix und dem ersten darzustellenden Punkt des eigentlichen Zeichens. Hierdurch ergibt sich der Abstand eines Zeichens vor dem vorangegangenen Zeichen. Das ist beim Amiga deshalb von Bedeutung, da die Zeichen am Bildschirm proportional dargestellt werden. Dadurch ist der Abstand der Zeichen voneinander — egal ob es sich um ein I oder M handelt — immer gleich. Das kerning kann als Variable bei der Zeichensatzdefinition angegeben werden (siehe font).

keyboard Die Tastatur des Amiga. Diese ist selbst ein (Mini-)Computersystem. Der eingebaute Prozessor — ein 6500/1 — läuft mit 1,5 MHz. Dazu kommen das »System ROM« mit 2 KByte, vier Ein/Ausgabeports für die Datenübertragung zum Amiga und 64 Byte RAM als Tastaturpuffer für etwa zehn Zeichencodes. Die Übertragung erfolgt mit 17 KBit/Sekunde. Da die Tastatur beim Einschalten einen Selbsttest durchführt, kann sie auch an einem bereits eingeschalteten Amiga angeschlossen werden. Die Ergebnisse des Selbsttests werden durch ein codiertes Blinken der Leuchtdiode an der Caps-Lock-Taste angezeigt. Liegt kein Fehler vor, so leuchtet diese Anzeige nur einmal kurz auf. Andernfalls haben die Blinkzeichen folgende Bedeutung: einmal — ROM Prüfsummenfehler; zweimal — RAM-Test-Fehler; dreimal — Tastaturtimer-Fehler; viermal — Kurzschluß in der Tastaturmatrix. Die Mengenangaben gelten pro Sekunde.

Die Tasten selbst sind mit Ausnahme der verschiedenen Steuertasten (SHIFT, ALT, AMIGA und CTRL) in einer 6 (Reihen) x 15 (Spalten)-Matrix angeordnet. Die Steuertasten werden über einen eigenen Port übertragen. Beim A1000 sind acht der 90 möglichen Tasten nicht benutzt, beim A500 und A2000 nur drei, weil die Tastatur dieser Computer erstens einen mit allen mathematischen Operatoren ausgerüsteten Zeichenblock und zweitens zwei zusätzliche Tasten (zwischen SHIFT links und <z> sowie rechts von RETURN) erhalten haben (zumindest in der europäischen Ausführung). Nicht in der Matrix enthalten ist die CAPS LOCK-Taste, die besonders verarbeitet wird.

Die von der Tastatur erzeugten Codes sind 7-Bit-Werte. Das achte Bit wird benutzt, um das Loslassen einer Taste zu signalisieren. Der alternative Zeichensatz wird mit der ALT- oder SHIFT-ALT-Tastenkombination erreicht. Die Shift-Tasten werden als »qualifiers« interpretiert, die eine eigene Codesequenz erzeugen. Dieser Sequenz wird bei der Verarbeitung der -> input events in Verbindung mit den -> keymaps der entsprechende Code der gedrückten Tastenkombination zugeordnet.

Das Anschlußkabel der A1000-Tastatur ist ein amerikanisches Telefonanschlußkabel mit einem Stecker vom Typ RJ11 und soll durch ein längeres Kabel desselben Typs ersetzbar sein. Berichten aus Amerika zufolge ist der Versuch, eine solche Verlängerung selbst zu machen, nicht zu empfehlen, da ein Kurzschluß beim Anbringen des Steckers verheerende Folgen für den Ein-/Ausgabebaustein (CIA 8520) des Amiga hat. Wegen des geänderten Anschlußsteckers und der erweiterten Tastaturbelegung ist der Anschluß der Tastatur des A2000 am A1000 nicht zu empfehlen.

keyboard.device Die Sammlung der Systemroutinen für die Verwaltung der Eingaben von der Tastatur (siehe console .device).

keymap (Tastaturbelegungsdatei). Diese Datei enthält die Definitionstabelle für die Umwandlung der »rohen« Steuersequenzen, die von der Tastatur kommen, in darstellbare Zeichen oder Steuercodes für die Bildschirmdarstellung.

Mit diesem Verfahren kann jeder Taste (mit Ausnahme der Shift-Tasten) eine bis zu sieben Zeichen lange (Steuer-)Sequenz zugeordnet werden. Zeichenketten mit einer Länge bis zu 32 Zeichen lassen sich auf einzelne Tasten legen (»hot keys«).

Die »Standardbelegung« (usa0) ist im ROM enthalten. Alle weiteren Belegungen müssen sich als entsprechende Datei im Verzeichnis »keymaps« befinden. Die Belegung »usa2« ist für das Dvorak-system (siehe setmap). Jedem Fenster kann eine andere keymap und damit ein anderer Zeichensatz zugeordnet werden.

keyboard shortcuts (Tastaturabkürzungen)
1. Die Benutzung der rechten Amiga-Taste in Verbindung mit einer alphanumerischen Taste, um Menüfunktionen zu wählen, ohne das Menü selbst abzurufen. In der Regel sind diese Abkürzungen programmspezifisch. Das Intuition Reference Manual enthält auf Seite 229 eine Vorschlagsliste mit Standardkombinationen für Workbench-Programme. Diese sind weitgehend den shortcuts des Notepad ähnlich.
2. Bei der Benutzung der linken Amiga-Taste zusammen mit einer alphanumerischen Taste wird eine bestimmte Aktion ausgelöst. Als Systemfunktionen sind die beiden Kombinationen <Amiga N> und <Amiga M> realisiert. Diese setzen die Workbench vor oder hinter den sichtbaren Screen. Leider hat diese Funktion einen Fehler. Sie funktioniert nur, wenn im Workbenchscreen ein Tool aktiviert wird. Man kann diesen Mangel umgehen, in dem in der Menüleiste die digitale Uhr aktiviert wird (die Zeile »RUN clock« gleich nach LOADWB in die startup-sequence einfügen).

keyword (Schlüsselwort, reserviertes Wort, Pflichtbefehl) Jedes Befehlswort einer Programmiersprache wird als Schlüsselwort bezeichnet. Solche Wörter können nicht als Variablenname verwendet werden. Innerhalb eines CLI-Befehls dienen Schlüsselwörter der eindeutigen Identifizierung eines Parameters (Beispiel: die Worte drive und name beim FORMAT-Befehl). Während die Parameter in der Regel an die im -> argument template beschriebene Reihenfolge gebunden sind, können mit Schlüsselwort eingeleitete Parameter an beliebiger Stelle stehen. Im template werden solche »Pflichtbefehle« mit dem Zusatz »/K« gekennzeichnet (siehe special characters).

Kickstart Disk (der »Tretstarter« des Amiga) Beim A1000 wird das -> ROM-Kernel von dieser Diskette in den Speicherbereich von $FC0000 bis $FFFFFF gelesen. Zum Zeitpunkt der ersten Auslieferung des Amiga 1000 war die Betriebssystemsoftware noch nicht fertig. Um die Ausstattung älterer Computer mit neuen Versionen des Kernels zu erleichtern, hat man es auf Diskette herausgegeben. Mit der Fertigstellung des Kernels in der Version 1.2 erhielten der A500 und A2000 ROM-Bausteine anstelle dieser Diskette (siehe bootstrap, WCS).

Kickstart Journal Eine seit der europäischen Einführung des Amiga zweimonatlich erscheinende technische Zeitschrift, die im Auftrag von Commodore Electronics von der englischen Software-Firma Ariadne (London) für die Verteilung an registrierte Software-Entwickler in der ganzen Welt ausgegeben wird. Einzelhefte kosten 5 Pfund, die ersten sechs werden als Sonderpaket für 25 Pfund angeboten (siehe developer's kit).

kludge (auch »hack« — zusammenschustern) Eine schnelle, aber nicht besonders elegante Methode eine Unzulänglichkeit (vor allem bei Hardware) zu beheben. Ein Beispiel dafür ist die Ausrüstung eines externen 5½-Zoll-Laufwerks mit einem Schalter, der die Diskettenwechselleitung simuliert.

L: ein »Library«-verzeichnis des Systems. Es enthält -> overlays für längere Befehlsroutinen sowie die nicht speicherresidenten Teile des Betriebssystems. Im Gegensatz zu den libraries im Verzeichnis -> LIBS: werden diese jedoch nicht mehr aus dem Speicher entfernt, wenn der Amiga sie nicht mehr braucht. Das Verzeichnis L: ist für die korrekte Funktion von Amiga-DOS unentbehrlich (siehe logical device).

label Der Name eines in Programmen enthaltenen Code- oder Datenabschnitts. Mit den Sprunganweisungen der jeweiligen Programmiersprache läßt sich die Programmausführung zu diesen Abschnitten lenken. Das Label eines Datenbereiches wird beim Zugriff auf diese Daten verwendet. Label wurden eingeführt, da alphanumerische Bezeichnungen einfacher zu merken sind als Speicheradressen. Manche Sprachenübersetzer oder -> Interpreter verlangen einen »:« am Ende des Labels.

layers.library eine Sammlung von Routinen, mit deren Hilfe -> Intuition die Verwaltung der -> Windows steuert. Ein »layer« ist eine Darstellungsebene des sichtbaren Bildschirms. Sie sollte aber nicht mit einer -> Bitplane verwechselt werden. Die Routinen der layers.library verwalten unter anderem die Sichtpriorität sich überlagernder Windows. Verdeckte Bereiche eines Window werden nach bestimmten Kriterien (-> refresh) nach ihrer Freilegung wiederhergestellt. Hierfür führen diese Routinen sogenannte »damage lists«, in denen Daten zu den verdeckten Bereichen abgelegt werden. Grundlage des Subsystems sind die sogenannten »clipping rectangles«, rechteckige Windowausschnitte, die verdeckten und sichtbaren Bereichen eines Window entsprechen.

library (Bibliothek) Im Betriebssystem des Amiga sind Programmroutinen eines Aufgabenbereiches in einem library genannten Modul zusammengefaßt. Eine solche Bibliothek ist eine Sammlung von Ansprungadressen, ein Datenbereich und schließlich die Routinen, auf welche die Ansprungadressen verweisen. Diese Bibliotheken können von allen Programmen im Multitaskingsystem genutzt werden. Man unterscheidet zwei Typen: residente und transiente libraries. Eine residente Bibliothek ist im Programmcode enthalten, der von der Kickstart geladen wird. Solche libraries »residieren« immer im Speicher. Transiente Bibliotheken befinden sich auf der Diskette im Verzeichnis -> LIBS: beziehungsweise L: und werden mit dem Aufruf (Öffnen) der library in den Speicher geladen.

Die Systembibliotheken können auch von Basic benutzt werden. Der Interpreter benötigt dazu die aus den -> .fd-files erzeugten -> .bmap-files. Bis zu fünf eröffnete Bibliotheken kann Amiga-Basic verwalten.

Höhere Programmiersprachen (C besonders intensiv) nutzen oft sogenannte Compilerlibraries. Hierdurch kann der Sprachumfang des Compilers relativ klein gehalten werden. Ein Aufruf einer nicht im Sprachumfang enthaltenen Funktion wird im Objektprogramm (siehe object code) entsprechend markiert und der nach dem Compilerlauf gestartete Linker (siehe link editor) versucht diese Funktionen in der Bibliothek zu finden.

Der Aufbau dieser Bibliotheken wird jedoch vom Hersteller des Compilers individuell bestimmt und unterscheidet sich somit von den Systemlibraries.

LIBS: das Systemverzeichnis für »open library«-Aufrufe. Es enthält die nicht speicherresidenten system libraries sowie vom Benutzer eventuell zusammengestellte Routinesammlungen (siehe logical device).

link editor (auch linker) Eine Hilfsroutine für Compiler und Assembler, mit der die erzeugten -> object code-Dateien zu einem lauffähigen -> load file (executable code) zusammengekoppelt werden. Der Linker bindet alle ausstehenden, externen Definitionen ein und paßt die relokierbaren Adressen dem endgültigen Format des load file an (siehe ALink, hunk). Externe Definition sind der Aufruf von Routinen oder Daten eines anderen Programmes oder einer Bibliothek.



2-Megabyte-Speichererweiterung für AMIGA 500 und AMIGA 1000

Die Karten sind für beide Rechner verwendbar

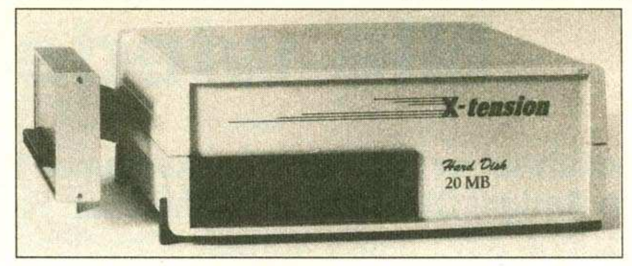
Version a: 2 Mega Byte bestückt
Version b: 2 Mega Byte bestückt, erweiterbar auf 4 Mega Byte
Erweiterung intern im CPU-Sockel einsteckbar

Einbinden ins Betriebssystem mit addmem
2 bzw. 4 Anfangsadressen einstellbar (ab 200000)
Aufgebaut mit 1-Megabit-Chips
9 Bit Hidden Refresh

GIGATRON ★ 4590 Cloppenburg
G. Preuth, R. Tiedeken
Resthauserstraße 128, Telefon 04471/3070
Dienstag und Donnerstag 15.00–18.30 Uhr



"Fast"-Disk



Blitzschnell durch Interleavefaktor 0

- z.B. AMIGA-"Assem" (ca. 64 KB) wird in weniger als 1,5 Sec. geladen.
- Anschlußfertig mit einer Speicherkapazität von 20 – 72 MB
- Einfache Software-Installation auf der Bootdiskette
- Demo-Diskette mit Installationsbeispielen
- ½ Jahr Garantie

Professionelles Arbeiten zu attraktiven Preisen:
20 MB DM 1498.–
40 MB DM 1998.–
72 MB DM 3498.–

Außerdem bei uns zu bestellen:
AMIGOS FÜR AMIGA™
Anschlußfertige Laufwerke: 3½" DM 339.–
5¼" DM 399.–

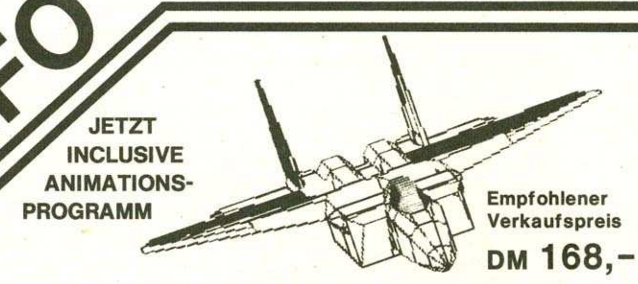


Wünschen Sie weitere Informationen rufen Sie uns an!
Tel. 0911/32 77 32 · Postfach 84 00 73 · 8500 Nürnberg 84

- Software
- Hardware
- Public Domain
- Systemberatung
- Diskettengroßhandel



Soft- und Hardware GmbH
Ihr *AMIGA*-Spezialist



Empfohlener Verkaufspreis
DM 168,–

Händleranfragen erwünscht.

3D CAD Programm für Commodore AMIGA*

SONDERAKTION !

Digi View Pal Version, in deutsch	DM **398,–**
incl. Adapter für 500er / 2000er	DM **438,–**
Adapter für 500er / 2000er	DM **48,–**
Update für Pal Version, in deutsch	DM **48,–**
D'Paint II oder D'Video 1.2	DM **198,–**
incl. deutsches Anwenderbuch	DM **228,–**
deutsche Anwenderbücher einzeln	DM **39,–**
XEBEC 20 MByte-Festplatte	DM **1248,–**
Butcher in deutsch!	DM **78,–**
Videoscape 3 D Pal Version, in deutsch	DM **388,–**
Page Flipper in deutsch	DM **98,–**

IHR DISKETTENGROSSHANDEL

3,5" Disketten DS/DD z.B. aus unserem „no name"-Angebot:
ab 10 Stück: DM **2,49** / Stück
ab 50 Stück: DM **2,45** / Stück
ab 100 Stück: DM **2,39** / Stück

Außerdem führen wir Marken-Disketten von SONY und dem Rolls Royce des Diskettenmarktes: MEMOREX.

Wir sind Profis in der Systemberatung
Beraten lassen kostet nichts!

Dunantstraße 53 · 5030 Hürth
☎ 0 22 33 / 4 10 81

Nie wieder Tippfehler!

Endlich ist er da, der Checksummer für den Amiga! Die Zeiten der endlosen Fehlersuche sind damit vorbei. »Checkie42« funktioniert für alle Programmiersprachen von Basic über C bis zum Assembler, wenn sie im ASCII-Format vorliegen. Er ist also universell einsetzbar und genügt allen Ansprüchen.

Damit wirklich jeder Leser unseren neuen Checksummer benutzen kann, wurde er in Amiga-Basic geschrieben. Trotz der sehr hohen Zuverlässigkeit und der strukturierten Programmierung konnte er recht kurz gehalten werden. Dabei bietet »Checkie42« aber allen nur erdenklichen Komfort und Sicherheit.

Nach dem Abtippen (siehe Listing), bei dem Sie diesmal noch gut aufpassen müssen, haben Sie endlich die erwünschte Hilfe beim Abtippen unserer Programme. Ab dieser Ausgabe werden alle Listings mit den Checksummen abgedruckt. Bevor wir zur Bedienung des Programms übergehen, soll zuvor noch erläutert werden, aus welchen Teilen sich eine Zeile in Zukunft zusammensetzt. Ein Beispiel für eine solche Zeile sieht so aus:

```
1 TTO print "Hallo!"
```

Die erste Zahl (»1«) stellt die Zeilennummer dar. Die Werte hierfür reichen von 1 bis maximal 9999. Nach einer Leerstelle folgt dann die eigentliche Prüfziffer, die sich noch mal unterteilt. Die ersten zwei Zeichen sind die Prüfziffern für den Text und die Zeilennummer. Erlaubt sind hier die Zeichen von »0« bis »9«, die großen (»A« bis »Z«) und die kleinen Buchstaben (»a« bis »z«). Durch die große Anzahl von verschiedenen Zeichen wurde erreicht, daß eine zweistellige Prüfsumme vollkommen ausreicht. Um dieselbe Fehlersicherheit mit Ziffern allein zu erreichen, würde man mindestens fünf bis sechs Stellen benötigen.

Das letzte Zeichen der Checksumme ist interessant, wenn die abgedruckten Listings strukturiert aufgebaut sind. Die dritte Stelle tritt nur dann auf, wenn sich der Beginn der Zeile verschoben hat. Deshalb erscheint sie auch nicht immer. Nach einer weiteren Leerstelle folgt dann endlich der eigentliche Text.

Doch nun zur Bedienung von »Checkie42«. Nachdem das Programm gestartet wurde, sehen Sie ein Fenster auf dem Bildschirm. In diesem Fenster erscheint zunächst eine Abfrage nach der gewünschten Datei. Hier können Sie einen neuen Dateinamen oder den Namen einer bereits bestehenden Datei eingeben. Zu letzterem Punkt kommen wir später.

Nach der Eingabe, die mit <RETURN> abgeschlossen wird, kommt nun das eigentliche Arbeitsfenster von »Checkie42« (siehe Bild). Links oben blinkt der Text »Checksumme:« und fordert Sie zur Eingabe der im Heft abgedruckten Prüfzeichen auf. Wenn Sie Wert darauf legen, daß das Listing auch bei Ihnen mit den richtigen Einrückungen generiert wird, müssen Sie auch die eventuell vorhandene dritte Stelle mit eingeben. Ansonsten geben Sie nur die ersten zwei Zeichen ein und drücken dann <RETURN>. Ist die dritte Stelle nicht vorhanden, beenden Sie die Eingabe auch mit <RETURN>.

Der Cursor steht nun zwischen den zwei Trennstrichen in der Mitte des Bildschirms. Davor hat das Programm automatisch die Zeilennummer geschrieben. Außerdem wurde der Cursor schon richtig positioniert, wenn die dritte Prüfziffer mit eingegeben wurde. Zum Beenden drücken Sie einfach die RETURN-Taste. Bei fehlerfreier Zeile wird der Raum zum Editieren gelöscht, die nächste Zeilennummer wird angezeigt und das Programm erwartet die Eingabe der nächsten Checksumme.

Leider wird es auch vorkommen, daß der Text einen Fehler enthält. Wenn Sie diesen entdeckt haben, können Sie den Cursor, der ganz am Anfang des Textes steht, an die entsprechende Stelle bewegen. Dabei verwenden Sie die Cursor-Steuertasten, um die Position zu erreichen. Mit der Backspace-Taste löschen Sie dann das Zeichen rechts vom Cursor. Auch das Einfügen von Zeichen ist ohne weiteres möglich. Dazu drücken Sie <TAB>, worauf an der Cursor-Position eine Leerstelle eingefügt wird. Dort können Sie nun das fehlende Zeichen eintragen. Für mehrere Zeichen muß entsprechend oft <TAB> gedrückt werden.

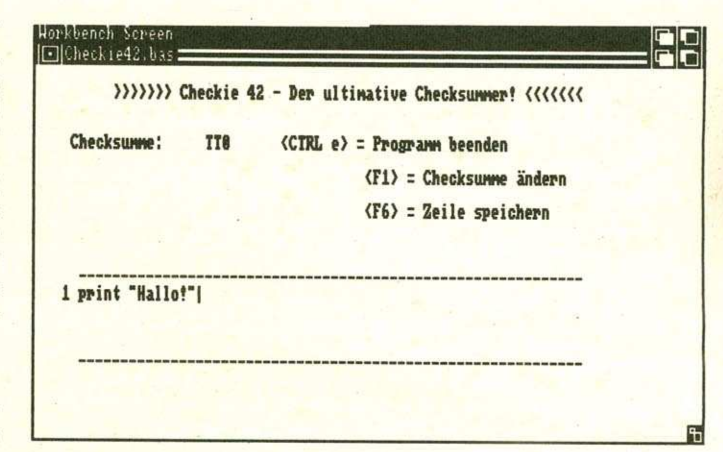


Bild. Das übersichtliche Fenster von »Checkie42«, in dem alle Eingaben gemacht und editiert werden können

Wenn Sie die Zeile trotz falscher Checksumme übernehmen wollen, drücken Sie die Taste <F6>. Dies ist zum Beispiel sinnvoll, wenn Sie Kommentare nicht mit eingeben wollen. Natürlich kann es auch vorkommen, daß Sie zwar die Zeile richtig, aber die Prüfsumme falsch eingegeben haben. In diesem Fall können Sie jederzeit die Taste <F1> drücken, worauf Sie die Checksumme ändern können.

Checkie42 speichert die einzelnen Zeilen nach Drücken von <RETURN> oder <F6> sofort in die Datei. Am Schluß des Listings müssen Sie dem Programm aber noch mitteilen, daß das Ende erreicht wurde. Das erreichen Sie durch Drücken von <CTRL E>. Wenn Sie Zeilen trotz falscher Prüfsumme übernommen haben, erfolgt eine Anzeige, wie viele Zeilen es waren. Längere Listings können Sie auch in Etappen abtippen. Wenn Sie keine Lust oder Zeit mehr haben, beenden Sie das Programm mit <CTRL E>. Bei dem nächsten Start geben Sie dann den gleichen Dateinamen ein und Checkie42 sucht automatisch diese Datei. Wenn sie gefunden wurde, lädt das Programm alle Zeilen und ermittelt so die nächste Zeilennummer. Ab dieser Position können Sie dann den Quelltext weiter eingeben.

Noch eine interessante Einzelheit von Checkie42 ist vielleicht für Sie von Interesse: Leerstellen vor und hinter dem eigentlichen Text werden nicht mitgeprüft. Wollten Sie also die Einrückung einer Zeile verändern, so stellt dies kein Problem dar.

Für diejenigen, denen die Basic-Version zu langsam sein sollte, noch ein kleiner Tip. Die Berechnung der eigentlichen Prüfsumme geschieht ab dem Label »CalcSumme:«. Dieser Teil ist sehr leicht in andere Programmiersprachen wie zum Beispiel C umsetzbar.

Nach dem Abtippen sollten Sie Checkie42 testen, indem Sie eines der in dieser Ausgabe abgedruckten Listings abtippen. Probieren Sie jede Funktion des Programmes durch, um etwaige Fehler durch das Abtippen zu entdecken. Wenn Ihr Checksummer dann richtig läuft, brauchen Sie in Zukunft keine Angst mehr vor Tippfehlern zu haben. Sie sparen damit viel Zeit für die vielleicht vergebliche Suche von tückischen Fehlern.

(Dieter Behlich/pa/rb)

Programmname:	Checkie42
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2

```
Programm : Checkie42
--------------------

 1 REM ***********************************
 2 REM *** Checksummer: Checkie 42       ***
 3 REM ***        Version 1.0            ***
 4 REM ***********************************
 5 Start:
 6   GOSUB Init
 7   GOSUB OpenDatei
 8   GOSUB Bild
 9 NeueZeile:
10   GOSUB Loeschen
11   GOSUB EingabeSumme
12   IF FEnde=wahr THEN Ende
13 Wiederholung:
14   GOSUB EingabeZeile
15   IF FEnde=wahr THEN Ende
16   GOSUB CalcSumme
17   IF FSumme=falsch THEN Wiederholung
18   GOSUB Uebernahme
19   GOTO NeueZeile
20 Ende:
21   GOSUB fertig
22   END
23
24 Init:
25   wahr=-1
26   falsch=0
27   LZeile=240  : REM Anzahl Zeichen/Zeile
28   LBZeile=60  : REM Anzahl Zeichen/Bildschrimzeile
29   AnzBZeilen=LZeile/LBZeile
30   zx=6 : zy=14   : REM Position Zeile
31   sx=20 : sy=5   : REM Position Checksumme
32   AnzCsZ=3       : REM Anzahl Ziffern/Checksumme
33   DIM z(LZeile)
34   DIM cs(AnzCsZ)
35   cs(AnzCsZ)=0   : REM Zeilenstart
36   a=0 : b=0 : c=0  : REM Hilfsvariablen
37   i=0 : j=0 : k=0  : REM Zählvariablen
38   READ Faktor(i)
39   WHILE Faktor(i)<>0
40     i=i+1
41     READ Faktor(i)  : REM Faktorenreihe
42   WEND
43   AnzFak=i
44   DATA 2,3,4,5,6,0
45   RETURN
46
47 OpenDatei:
48   ON ERROR GOTO Dateifehler
49   CLS
50   LOCATE 3,5
51   dn$="":INPUT "Dateiname: ";dn$
52   Zeile=1
53   OPEN dn$ FOR INPUT AS #1
54   IF FFehler=53 THEN NeueDatei
55   WHILE NOT EOF(1)
56     LINE INPUT#1,e$
57 REM PRINT e$
58     Zeile=Zeile+1
59   WEND
60   CLOSE 1
61 NeueDatei:
62   OPEN dn$ FOR APPEND AS #1
63   RETURN
64 Dateifehler:
65   FFehler=ERR
66   IF FFehler=53 THEN RESUME NEXT
67   ON ERROR GOTO 0
68
69 Bild:
70   CLS
71   LOCATE 2,10
72   PRINT ">>>>>>> Checkie 42 - Der ultimative Checksummer!
     <<<<<<<"
73   LOCATE 13,zx
74   FOR i=1 TO 2
75     FOR j=1 TO LBZeile
76       PRINT "-";
77     NEXT j
78     LOCATE 13+AnzBZeilen+1,zx
79   NEXT i
80   LOCATE 5,30
81   PRINT "<CTRL e> = Programm beenden"
82   LOCATE 7,40
83   PRINT "<F1> = Checksumme ändern"
84   LOCATE 9,40
85   PRINT "<F6> = Zeile speichern"
86   RETURN
87
```

Listing. Der ASCII-Checksummer »Checkie42« verhindert Fehler beim Abtippen von Listings. Bitte ohne Zeilennummern eingeben.

Fortsetzung auf Seite 80

Es hat wenig Sinn,
Werbung zu machen.
Man kauft die Ware dort,
wo sie am günstigsten ist.
Zumal wir 1 Jahr Garantie auf
elektrische Geräte gewähren.

Fa. Dirk Böhme
Soft & Hardware

Panoramastraße 16,
7057 Leutenbach 2
Tel. 07195/62509
Rund um die Uhr

Joysoft
Inh. Gabriele Hartmann

laut Umfrage einer deutschen Software-Zeitschrift sind wir

DEUTSCHLANDS BELIEBTESTES SOFTWAREHAUS MIT DEM BESTEN SERVICE
UND DAS BEWEISEN WIR TÄGLICH

24 Std. Bestell-Annahme
24 Std. Eil-Lieferservice auf Anfrage
Eigene Lagerhaltung, deshalb prompte Lieferung

AMIGA BARDS TALE I 89,—, PHANTASIE I 79,—, GARRISON 59,90, DER HAUCH DES TODES 54,90, FLIGHTSIMULATOR II 139,—, KAMPFGRUPPE 69,—, MINDBREAKER 29,90, PAC BOY 29,90, ROADWAR 2000 54,90, ROCKET ATTACK 29,90, SHOOTING STAR 29,90, SPACE FIGHT 29,90, THE FINAL TRIP 29,90, TYPHOON 69,—, VADER 29,90

Laden und Versand:	Laden Köln 1:	Laden Düsseldorf:
Berrenrather Str. 159	Matthiasstr. 24-26	Humboldstr. 84
5000 Köln 41	**5000 Köln 1**	**4000 Düsseldorf 1**
Tel.: (0221) 416634	Tel.: (0221) 239526	Tel.: (0211) 6801403

ODER TELEFONISCH BESTELLEN UNTER
0221 – 416634 10 - 18.30 Uhr
0221 – 425566 24-Std. Service

AMIGA-SOFTWARE

Public-Domain-Disketten

Fish Disk	**1 bis 92**
Auge 4000	**1 bis 11**
Faug	**1 bis 39**
Panorama	**1 bis 40**

Inhalt:
Spiele, Bilder, Video-Clips, Musik, Anwendungen, Programmiersprachen, Textverarbeitung, DFÜ und Demos der bekanntesten TOP-Programme

Jede PD-Diskette 6,50 DM

Aktuelle Spiele auf Anfrage

A. Fischer – 05257/4347
4794 Hövelhof

Minipreise für Amiga-Laufwerke

1 MB 3½″ FD 35 FN Teak Metallgehäuse, anschlußfertig	290 DM
1 MB 5¼″ FD 55 FR Teak, Metallgehäuse, anschlußfertig	375 DM
1 MB 3½″ FD 35 FN Teak	270 DM
1 MB 5¼″ FD 55 FR Teak	290 DM
1 MB 3½″ NEC 1036 A	290 DM

Angelika Huber · Elektr. Bedarf
Wörnitzstraße 3
8850 Donauwörth
Telefon 0906/5567

AMIGA-SOFTWARE ZU SUPERPREISEN

Fred-Fish ....**Nr. 1-92** FAUG ....**Nr. 1-39**
PANORAMA........**Nr. 1-48** Amicus..............**Nr. 1-16**
u.a. (Katalogdisk mit Kurzbeschreibung gegen 5.- DM anfordern)
Einzeldisk......**5.00 je Disk** ab 10 St........**4.80 je Disk**
ab 20 St..........**4.50 je Disk** ab 40 St........**4.30 je Disk**
alle Preise incl. 2 DD Diskette

TORNADOS · Die Super PD-Serie. Alle Prg. laufen garantiert auf Amiga 500, 1000 u. 2000.
Nr. 1-30 **6.00 je Disk**
BOOTWRITER · Farbige Endlos-Laufschrift mit Hintergrundscrolling binden Sie problemlos im Bootblock ein. **17.- DM**
IFF-CON · Werden Sie zum Graphikzauberer. Umwandlung von Bildern in Sprites und vieles mehr. **27.- DM**
Inland: Porto + Verp. je Bestellung 3.- DM
(nicht bei Anforderung v. Infos o. Katalogdisk)
Ausland: Porto + Verp. je Bestellung 6.- DM
(nicht bei Anforderung v. Infos o. Katalogdisk)
Lieferung gegen Vorkasse oder V-Scheck
Achtung! Kein Ladenverkauf

Bestellung und Anfragen an:
PD-Shop
Opladener Straße 30, D-4018 Langenfeld

---

1 MByte-RAM-Karte für alle Bus-Systeme, z.B. VME, EC8 steckerfertig, 100x160 mm .... **548,— DM**
2 MByte-RAM-Karte für KWS 235x160 mm, steckerfertig .... **1098,— DM**
DIN-A3-Plotter CP64 für C64 .... **875,— DM**
CP65 für KWS oder andere Systeme (V.24/Centronics) HP-GL-kompatible, 150 mm/sec.
Auflösung 0,05 mm .... **1111,— DM**
Komplett für KWS nur .... **1999,— DM**

41256-120	**6,40**	TTL-HC	
511000-12	**55,—**	00, 02, 04, 08,	
41464-120	**7,80**	10, 11, 20, 21	**0,60**
2764-250	**6,40**	73, 74, 86	**0,95**
27256-250	**9,80**	245	**2,20**
325572	**39,90**	HCT 04	**0,60**
ICL7109	**24,95**	137, 240, 241,	
MK50395N	**36,90**	244, 373, 374	**1,60**
MK50398N	**35,90**	C-MOS 4000, 01	
MAX232	**13,95**	02, 07, 11, 12	**0,45**
ICL7106	**9,40**	7805-7824	**0,95**
ICL7107	**9,40**	IC-Sockel	
XR2206	**8,—**	Low-cost pP	**0,018**
XR2207	**8,20**	IC-Sockel	
74LS03,10	**0,37**	Präz. pP	**0,04**

Händleranfragen erwünscht

Martin Fleitmann electronic
Albert-Schweitzer-Weg 12 · 4600 Dortmund 18
Telefon: 0231/673868

ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG ! ACHTUNG



An alle AMIGA Programmierer !



Wir suchen **AMIGA** Programme aus allen Anwendungsgebieten, für die Vermarktung im gesamten Bundesgebiet. Wir übernehmen die Kosten für den Vertrieb (Werbung, Verpackung, Versand usw.).

Wir bieten Ihnen eine Umsatzbeteiligung bis zu

50 %

Wenn Sie Interesse haben, wenden Sie sich noch heute an uns:

SOFTWARE 2000



A. Wardenga
Lange Str. 51
2320 PLOEN



Bekanntmachung!!

X-tension Harddisks
Made in Germany

20 MB	1598.–
40 MB	2098.–
72 MB	3475.–

Superschnell
durch Interleave 0
Schaltnetzteil u.v.m.

Fa. T. Eder & P. Archinger
Peyerstraße 35
8500 Nürnberg 80
0911/329558 17–20 Uhr

ALCOMP
COMPUTERHARDWARE

AMIGA: Speichererweiterung

für 512 K zusätzliches RAM (Echtzeituhr nachrüstbar), komplett mit 512 K:	**189,–**
Leerplatine + Stecker für AMIGA 500:	**39,–**
Uhrenchip 6242	**24,–**

BESTELLUNG/VERSAND

ALCOMP · A. Lanfermann
Lessingstr. 46 · 5012 Bedburg · Tel. 02272/1580

Nachnahmeversand, NN-Spesen 7,50 DM, bei Vorkasse 3,– DM. Auslandsbestellungen nur gegen Vorkasse + 15,– DM Versandkosten. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikgewerbes.



Amiga Speichererweiterung intern, Fastram
1 MB **749,—** 2 MB **1248,—** 4 MB **2048,—**
(Test A-Special 10/87)

Laufwerk 1036A, abschaltbar, anschlußfertig im amigafarbenen Blechgehäuse **329,—**
(Test Kickstart 10/87)

Amiga 500, 512 K abschaltbar	**249,—**
Amiga 500, 2 MB abschaltbar	**889,—**
Amiga Digitizer A500, 1000, 2000	**99,—**
Amiga Genlock-Interface	**489,—**

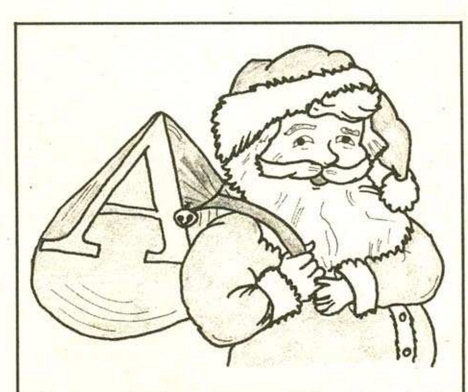
Coll-Card und Coll-Prom, die ersten Epromkarten und Eprombrenner für den Amiga

Coll-Card A500 + A1000, 512 K best.		**298,—**
weitere 512 K-Karte 49,—	2 MB	**419,—**
Coll-Card A2000	2 MB	**398,—**
Coll-Prom A500 + A1000		**398,—**
Coll-Prom A2000		**650,—**

Weiteres Zubehör (5¼"-Laufwerk, A500, Soft) auf Anfrage lieferbar. Anfragen ab 18.00 Uhr unter Telefon 06163/1278.

FUTURE VISION

Friedrich-Veith-Str. 21, 6128 Höchst/Odw.



Und was steht auf Ihrem Wunschzettel?



B.Ludewig + Th.Wittwer GbR
Bielefelder Str.121 · 4802 Halle (05201/7555

INTERSOFT

Nohlstraße 76 · D-4200 Oberhausen 1 · Telefon: 0208 / 80 90 14

LaSch, das Buch und Software Haus

Inh.: Rainer Langner & Franz Schnitzler, GbR

Go Amiga Datei	**DM 199,—**
Demoversion	**DM 19,—**
Superbase	**DM 229,—**
Demoversion	**DM 19,—**
Der Hauch des Todes (007)	**DM 59,—**
The Guild of Thieves	**DM 79,—**
Gnome Ranger	**DM 49,—**

Public-Domain-Pool

über 400 Disketten für **Amiga**
ab 5,— DM inkl. Diskette!
· Fred Fish · Panorama · Faug · Amicus · Auge · Slide-Shows · Software Demos · Kick Star
Katalog gegen 5,— DM (V-Scheck/Briefmarken)
bitte Computertyp angeben!

AMIGA 500	1148,—
AMIGA 2000	2398,—
AMIGA 2000 mit 1081	3048,—
AMIGA Colormonitor 1081	698,—
AMIGA 1010 2. Floppy 3.5	469,—
2000 2. Floppy intern	379,—
2000 PC/AT Karte	1898,—
Commodore MPS-1200	529,—
Epson LX-800	598,—
Star NL-10 inkl. Interf.	598,—
Panasonic KX-P 1082	749,—
NEC P6 bzw. MPS-2000	1198,—
NEC-P6 color/MPS-2000C	1495,—
OKIMATE 20 mit Interface nach Wahl	448,—
Wiesem. 92000/G	129,—
Amiga Software PC-Emulator für Amiga 1000	99,—
Superbase für Amiga	249,—
Logistix für Amiga	359,—
Deluxe Paint II	279,—
DATAMAT/TEXTOMAT Amiga	99,—
GEOS Programm für C 64	59,—
Grünmonitor VC-1900	199,—
ATARI 1040 STF komplett mit Monitor SM124/Maus/TOS	1498,—
VC 1541 C	348,—
Vortex Drivecard 20 MB f. PCs + Amiga 2000 (PC-Seite)	998,—

Electronic-Bauteile/Data-Becker/Markt & Technik usw.
Bitte fragen Sie an!

Telefon 07541/73122
Versand und Ladenverkauf
REICH-ELECTRONIC
Allinger Str. 86/1, 7990 Friedrichshafen

PUBLIC DOMAIN SERVICE RUHRSOFT

Weit über 250 Disketten lieferbar!
Aktuell im Schnellversand!

Einzelstück	**8,00 DM**
ab 10 Stück	**je 7,00 DM**
ab 20 Stück	**je 6,00 DM**
ab 30 Stück	**je 5,00 DM**
ab 50 Stück	**je 4,50 DM**

Preise incl. 2DD 3½"-Disks.

Super aktuell: FRED FISH bis Nr. 102 — **Ab sofort auch erhältlich: Gold-Fish-Serie**

Katalogdiskette gegen 5,- DM anfordern, Betrag wird bei Serienbest. angerechnet! Einzigartig: Wir liefern auch auf 5¼"-Disketten, hierbei alle Preise —1 DM, z.B. ab 50 Stck. je 3,50 DM!

Markus Scheer · Kapellenweg 42
4630 Bochum 5 · Tel. 0234/411958

Amiga Laufwerk NEC 1036 A	**2 MB RAM Erweiterung**
für Amiga 500/ 1000/2000	autokonfig,
3,5 Zoll, slimline	abschaltbar,
(10,5x17,5x3,5 cm)	durchgeführter Bus,
amigafarbener Strukturlack Einbrennlack.,	amigafarbenes Metallgehäuse
Metallgehäuse anschlußfertig	100% Software-kompatibel
349,— DM	**929,— DM**

SWS Computersysteme G.d.b.R.
Beratung — Planung — Verkauf — Service
Am Rathaus 8, 8395 Hauzenberg
Telefon 08586-2174

A&O

Analyse und Optimierung elektrischer Netzwerke

Deutsche Version für alle AMIGA-Konfigurationen ab 512k

Analyse: Simulation beliebiger elektrischer Schaltungen aus aktiven und passiven Komponenten. Berechnung der Netzwerkfunktionen U_2/U_1, S_{11}, S_{12}, S_{21}, S_{22}, Z_{ein} und Z_{aus} nach Betrag und Phase.

Optimierung: Bei vorgegebenen Schaltungseigenschaften werden die optimalen Bauelementwerte berechnet (CAD).

- Eine komfortable Benutzeroberfläche mit vielen Menüs und mehreren Fenstern stellt die einfache Programmbedienung sicher.
- Es stehen verschiedene Grafikoptionen für lineare und logarithmische Darstellungen der Ergebnisse zur Verfügung.
- Große Leistungsfähigkeit und kurze Rechenzeiten durch effektive C-Implementierung.

Der Endpreis von DM 265,- schließt die Programmpflege über 12 Monate ein. Die Lieferung erfolgt per Nachnahme.
Weitere Programminformationen auf Anfrage.

Dr. U. Christ Klawitterstr. 27, 2800 Bremen 61 Telefon (0421) 825487

Musik- und Grafiksoftware Shop
Das Spezialgeschäft für Grafiksoft- und Hardware
Wasserburger Landstr. 244 ★ 8000 München 82
Telefon 089/4306207

SCANNER ATARI ST und AMIGA IBM (DIN A4, 200 Dots/Inch)

Flachbett-Scanner, 10 Sek. Scannzeit, mit eingebautem Thermodrucker. Verwendung als Scanner, Kopierer und Hochgeschwindigkeitsdrucker. Mit integriertem Zeichenprogramm für hochauflösende Bilder. Kompatibel zu Degas, MonoStar, Campus, Fleetstreet Publisher, Publishing Partner und vielen anderen Zeichen- und DTP-Programmen, Schrifterkennung und Telefax-Software in Vorbereitung **2998,-**

Sound-Digitizer für ATARI ST und AMIGA
Digitalisier-Tableau ATARI ST und IBM
Videodigitizer (ATARI/AMIGA/IBM PC AT/C64)
Zahlreiche Grafik- und DTP-Programme für Atari ST und Amiga auf Lager

DIGI-PAINT Das neue 4096-Farben Malprogramm für AMIGA
DELUXE-PAINT II PAL-Version mit deutschem Manual

Fordern Sie unseren kostenlosen Katalog an. Ausführliche INFO zu allen Produkten. Täglicher Versand per Nachnahme oder Vorkasse!

Rufen Sie uns einfach an oder besuchen Sie uns in unserem Laden!
MO-FR 10—18.30 UHR SA 9—13.00 UHR

```
88 Loeschen:
89   LOCATE zy,1
90   FOR i=1 TO AnzBZeilen
91     PRINT TAB(zx);SPACE$(LBZeile)
92   NEXT i
93   FOR i=0 TO LZeile
94     z(i)=32
95   NEXT i
96   LOCATE zy,1
97   PRINT USING "####";Zeile
98   aPos=0
99   RETURN
100
101 EingabeSumme:
102   Farbe=0 : i=1
103   LOCATE sy,sx+1
104   PRINT "..."
105 blinken:
106   Farbe=Farbe XOR 1
107   COLOR Farbe,0
108   LOCATE sy,sx-15
109   PRINT "Checksumme:"
110   e$=INKEY$
111   IF e$="" THEN blinken
112   e=ASC(e$)-48
113   IF e=-43 THEN
114     FEnde=wahr   : REM <CTRL e>
115     GOTO ESEnde
116   ELSE
117     IF e<>-35 THEN keinEnde   : REM <CR>
118     IF i<AnzCsZ THEN blinken
119     GOTO ESEnde
120   keinEnde:
121     IF e<0 THEN blinken
122     IF e>9 THEN e=e-7
123     IF e<0 THEN blinken
124     IF e>35 THEN e=e-6
125     IF e<0 THEN blinken
126     IF e>61 THEN blinken
127     LOCATE sy,sx+i
128     COLOR 1,0
129     PRINT e$;
130     cs(i)=e
131     i=i+1
132     IF i<=AnzCsZ THEN blinken
133   END IF
134 ESEnde:
135   COLOR 1,0
136   LOCATE sy,sx-15
137   PRINT "Checksumme:"
138   RETURN
139
140 EingabeZeile:
141   x=cs(AnzCsZ)
142 weiter:
143   LOCATE zy+INT(x/LBZeile),zx+(x MOD LBZeile)
144   IF x>aPos THEN aPos=x
145   e$=INPUT$(1)
146   e=ASC(e$)
147   IF (e AND 127)<32 THEN Controlcode
148   PRINT e$
149   z(x)=e : e=30
150 Controlcode:
151   IF e=13 THEN
152     RETURN
153   ELSEIF e=30TTHEN
154     a=1
155   ELSEIF e=29 THEN
156     a=LBZeile
157   ELSEIF e=31 THEN
158     a=-1
159   ELSEIF e=28 THEN
160     a=-LBZeile
161   ELSE
162     GOTO noCr
163   END IF
164   x=x+a
165   IF x>=0 AND x<LZeile THEN weiter
166   x=x-a
167   GOTO weiter
168 noCrs:
169   IF e=8 THEN
170     FOR i=x TO aPos
171       z(i)=z(i+1)
172       PRINT CHR$(z(i));
173      IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
174     NEXT i
175     z(aPos)=32 : PRINT " "
176     aPos=aPos-1
177   ELSEIF e=9 THEN
178     IF aPos>x THEN
179       FOR i=aPos TO x STEP -1
180         z(i+1)=z(i)
181       NEXT i
182       z(x)=32
183       aPos=aPos+1
184       IF aPos=LZeile THEN aPos=aPos-1:z(LZeile)=32
185       FOR i=x TO aPos
186         PRINT CHR$(z(i));
187         IF i MOD LBZeile=59 THEN PRINT:PRINT TAB(zx);
188       NEXT i
189     END IF
190   ELSEIF e=129 THEN
191     GOSUB EingabeSumme
192     x=cs(AnzCsZ)
193   ELSEIF e=134  THEN
194     RETURN
195   ELSEIF e=5 THEN
196     FEnde=wahr
197     RETURN
198   END IF
199   GOTO weiter
200
201 CalcSumme:
202   a=0 : b=0 : c=0
203   IF e=134 THEN
204     FSumme=wahr
205     FF6=FF6+1
206   ELSE
207     WHILE z(aPos)=32 AND aPos>0
208       aPos=aPos-1
209     WEND
210     IF aPos>0 THEN
211       WHILE z(c)=32
212         c=c+1
213       WEND
214     END IF
215     FOR i=c TO aPos
216       j=(i-c) MOD AnzFak
217       k=(i+1-c) MOD AnzFak
218       a=a+((z(i) AND 127)-32)*Faktor(j)
219       b=b+((z(i) AND 127)-32)*Faktor(k)
220     NEXT i
221     FSumme=(cs(1)=(a+Zeile) MOD 62) AND (cs(2)=(b+Zeile) MOD 62)
222   END IF
223   RETURN
224
225 Uebernahme:
226   FOR i=0 TO aPos
227     PRINT#1,CHR$(z(i));
228   NEXT i
229     PRINT#1,""
230   Zeile=Zeile+1
231   RETURN
232
233 fertig:
234   CLOSE 1
235   CLS
236   LOCATE 12,35
237   PRINT "F E R T I G !!!"
238   LOCATE 20,1
239   IF FF6<>0 THEN
240     PRINT "ACHTUNG!!!  ";
241     PRINT FF6;" Zeile(n) wurde(n) ungeprüft gespeichert."
242   END IF
243   RETURN
(C) 1987 M&T
```

Listing. Der ASCII-Checksummer »Checkie 42« verhindert Fehler beim Abtippen von Listings (Schluß)

Die schnellste Hardcopy

Das Prinzip der Druckertreiber für den Amiga ist sicher sehr gut, aber die Geschwindigkeit und Qualität des Ausdrucks lassen zu wünschen übrig. Mit »Speedhardcopy« sind schnelle Schwarzweißausdrucke kein Wunschtraum mehr.

Nicht jeder hat einen Farbdrucker und bei vielen Ausdrucken ist auch keine Farbe nötig. Trotzdem dauert eine Hardcopy auch in Schwarzweiß sehr lange. Um die Geschwindigkeit zu steigern bleibt nur ein Ausweg: Die Daten werden nicht über den Druckertreiber an den Drucker geschickt, sondern direkt. Dazu muß natürlich das Programm selbst die Umwandlung der Grafik vornehmen und dieses nicht an »prt:« schicken, sondern einfach an die parallele Schnittstelle (par:).

Dies bringt leider mit sich, daß »Speedhardcopy« für jeden Drucker angepaßt werden muß. Dies ist jedoch nicht sonderlich schwierig, da das Programm in C geschrieben ist. Das Listing ist für den NEC P6/P7 geschrieben, läuft aber natürlich auch mit der Farbversion dieses Druckers. Die notwendigen Änderungen für andere Drucker werden später noch erklärt.

Der NEC P6 beherrscht elf verschiedene Grafikmodi, die auch von Speedhardcopy benutzt werden können. Durch die Ausnutzung der 24 Nadeln erhält man eine wesentlich bessere Qualität als sonst.

Nach Starten des Programms von der Workbench oder dem CLI erscheint ein Fenster, in dem die 11 Druckmodi des NEC P6 aufgelistet sind. Hierbei steht DPI für die Anzahl der Punkte pro Inch. Die besten Resultate sind natürlich bei der höchsten Auflösung zu erhalten.

Nun muß noch die Eingabe der gewünschten Bitplane erfolgen. Dies ist im Normalfall die Ebene 0, da dann die Hauptvordergrundfarbe ausgedruckt wird. Bei nicht veränderten Farben ist dies beim Workbenchscreen die Farbe Weiß.

Mit den folgenden Eingaben kann die Breite und Höhe des Ausdrucks beinflußt werden. Dies ist sinnvoll, da in der höchsten Auflösung der Ausdruck sonst nur 4,5 mal 3,5 Zentimeter groß wäre. Die Angabe in Klammern gibt den Wert an, mit dem das Bild genau in die Breite eines DIN-A4-Blattes paßt. Die Parameter für Höhe und Breite müssen gleichgroß gewählt werden, wenn keine Verzerrung gewünscht wird.

Gleich nach der letzten Eingabe beginnt der Druckvorgang. Deshalb sollte man sich vorher noch vergewissern, daß am Drucker alles in Ordnung ist. Falls der Drucker nicht eingeschaltet sein sollte, bricht das Programm selbständig ab.

Auch für andere Drucker

Nun zu den wichtigen Punkten von Speedhardcopy, die man kennen muß, um das Programm für andere Drucker anzupassen.

»MAX_BREITE« und »MAX_HOEHE« gibt die maximale Zahl der Druckpunkte an, um die die Hardcopy in der Breite beziehungsweise Höhe vergrößert werden kann. In der Praxis werden aber niemals größere Werte sinnvoll sein, als die hier eingestellten. Auch die Werte »MAX_XAUFL« und »MAX_YAUFL« bedürfen keiner Änderung, da sie die maximale Auflösung eines Bildschirms angeben.

Nun folgen einige Werte, die Sie in Ihrem Druckerhandbuch nachschlagen müssen. Mit der Variablen »GraphicAbstand« legen Sie Escape-Sequenz fest, mit der der richtige Zeilenabstand für Grafik festgelegt wird.

In der nächsten Zeile folgt mit der Definition von »NormalAbstand« die Bestimmung der Sequenz, die wieder auf normalen Zeilenabstand schaltet.

Das Feld »InitGraphic« besteht aus fünf Bytes, die den Grafikmodus des Druckers einschalten. Auch hier müssen Sie die passenden Werte aus Ihrem Druckerhandbuch eintragen.

Im Feld »MaxPoints« werden die maximal möglichen Druckpunkte pro Druckzeile festgelegt.

Die einzige Routine, die eventuell geändert werden muß, ist »Print()«. In dieser Funktion wird die Zeile »PrtLine«, in der die Grafikdaten für eine Druckerzeile stehen, ausgegeben. Falls die Daten spaltenweise an den Drucker geschickt werden müssen, was bei fast jedem Drucker der Fall ist, müssen die einzelnen Druckzeilen erst konstruiert werden. Für diese Aufgabe wird das Feld »PrtBuffer« benutzt.

Speedhardcopy wurde für den Lattice-C-Compiler geschrieben, läßt sich aber auch mit dem Aztec-Compiler übersetzen.

Die Aufrufe für den Lattice-Compiler sehen so aus:

```
lc speedhardcopy
blink lib:c.o,speedhardcopy.o lib lib:lc.lib,lib:
amiga.lib
```

Besitzen Sie den Aztec-Compiler verwenden Sie bitte folgende Aufrufe:

```
cc speedhardcopy +l -s
ln speedhardcopy.o -lc32
```

Es ist auch möglich das Programm so umzuschreiben, daß ein Farbdrucker angesteuert wird. Hierauf wurde aber bewußt verzichtet, da die Druckgeschwindigkeit wieder sinken würde. Trotz der hohen Qualität benötigt das Programm zum Beispiel für eine Hardcopy im Modus 11, also der höchsten Auflösung, und vier Punkten für Breite und Höhe nur noch ungefähr 5 Minuten. Für viele Modi liegt die Druckzeit sogar deutlich unter einer Minute!

Es ist also endlich vorbei mit dem ewigen Warten auf einen Ausdruck. (Timo Kissel/rb)

Programmname:	Speedhardcopy
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40 oder Lattice-C V3.10
Aufrufe:	siehe Text

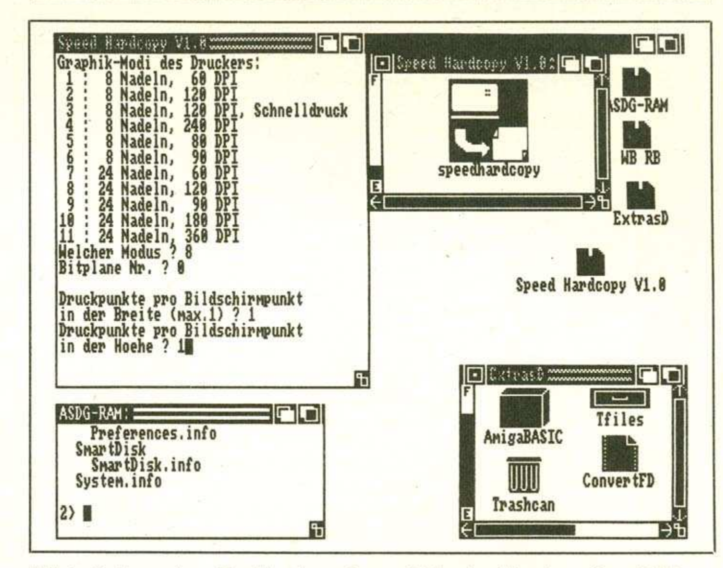


Bild. Sehr schnelle Hardcopies mit fantastischer Qualität sind mit Speedhardcopy kein Wunschtraum mehr

```
 1 L40  /***********************
 2 oS   *  SPEED HARDCOPY V1.0 *
 3 3c   *  für NEC P6/P7       *
 4 DA   *   von Timo Kissel    *
 5 eN   ***********************/
 6 px   #include "exec/types.h"
 7 wL   #include "stdio.h"
 8 e7   #include "intuition/intuition.h"
 9 02   #include "intuition/intuitionbase.h"
10 BN   #include "libraries/dosextens.h"
```

Listing. Der C-Quellcode von »Speedhardcopy«. Bitte mit dem Checksummer (siehe Seite 76) eingeben.

```
11 7U  #include "devices/parallel.h"
12 di  #define MAX_BREITE 8
13 GT  #define MAX_HOEHE 8
14 IW  #define MAX_XAUFL 640
15 4W  #define MAX_YAUFL 512
16 fc  #define MAX_BUFFER 100
17 9m  struct IOExtPar IORpar;
18 1Y  struct MsgPort *HCPort;
19 jL  struct IntuitionBase *IntBase;
20 fB  struct FileHandle *HCWindow;
21 KM  UBYTE GraphicAbstand[3] = { 27,65,8 };
22 RV  UBYTE NormalAbstand[2] = { 27,50 };
23 6q  UBYTE InitGraphic[5] = { 27,42,1,128,2 };
24 9U  UBYTE GModes[11] = { 0,1,2,3,4,6,32,33,38,39,40 };
25 f6  UBYTE LineFeed = { 10 };
26 92  ULONG MaxPoints[11] = { 476,953,953,1906,635,714,476,953,714,
       1430,2859 };
27 fi  /* ULONG MaxPoints[11] = { 816,1632,1632,3264,1088,1224,816,1
       632,1224,2448,4896 };
28 fB3   fuer NEC P7 */
29 vs0 UBYTE AktBit,AktByte,VertBytes;
30 6g  UBYTE PrtLine[MAX_BREITE*MAX_XAUFL];
31 Lo  UBYTE PrtBuffer[MAX_BREITE*MAX_XAUFL*3];
32 GG  UBYTE *BitplaneAdr;
33 VA  USHORT Breite,Hoehe,AnzSpalten,MemNeeded;
34 q2  char Buffer[MAX_BUFFER+1];
35 G2  extern struct MsgPort *CreatePort();
36 bG  void WritePrt();
37 TE  void Print();
38 sJ  void WriteScr();
39 Fa  ULONG ReadScr();
40 8Q  void main()
41 b4  {
42 KB3   ULONG GMode,Error,Max,WhichPlane;
43 D7    UBYTE YFaktor;
44 Iv    register UBYTE z,XFaktor,h;
45 5P    register UBYTE *a;
46 xi    register USHORT x,y,r1;
47 Q2    struct Screen *HCScreen;
48 xd    struct BitMap *HCBitmap;
49 T0    AktByte = 0;
50 RD    AktBit = 128;
51 Yg    XFaktor = 1;
52 bk    YFaktor = 1;
53 aT    VertBytes = 1;
54 RA    if ((IntBase = (struct IntuitionBase *)OpenLibrary("intuit
         ion.library",0)) == 0)
55 pI    {
56 1j6     printf("Can't open intuition.library !\n");
57 RN      exit(99);
58 wR3   }
59 Yw    sprintf(Buffer,"CON:0/0/320/180/Speed Hardcopy V1.0");
60 xu    if ((HCWindow = (struct FileHandle *)Open(Buffer,MODE_NEWF
         ILE)) == 0)
61 v0    {
62 4z6     printf("Can't open Window !\n");
63 FD      CloseLibrary(IntBase);
64 YU      exit(99);
65 3Y3   }
66 1V    WriteScr("Graphik-Modi des Druckers:\n 1 :  8 Nadeln,  60
         DPI\n 2 :  8 Nadeln, 120 DPI\n");
67 mz    WriteScr(" 3 :  8 Nadeln, 120 DPI, Schnelldruck\n 4 :  8 N
         adeln, 240 DPI\n 5 :  8 Nadeln,  80 DPI\n");
68 Fi    WriteScr(" 6 :  8 Nadeln,  90 DPI\n 7 : 24 Nadeln,  60 DPI
         \n 8 : 24 Nadeln, 120 DPI\n");
69 ee    WriteScr(" 9 : 24 Nadeln,  90 DPI\n10 : 24 Nadeln, 180 DPI
         \n11 : 24 Nadeln, 360 DPI\nWelcher Modus ? ");
70 IS    if ((GMode = ReadScr()) >= 7) VertBytes = 3;
71 8r    InitGraphic[2] = GModes[--GMode];
72 ms    WriteScr("Bitplane Nr. ? ");
73 TH    WhichPlane = ReadScr();
74 nN    HCScreen = IntBase->ActiveScreen;
75 Lg    HCBitmap = &HCScreen->BitMap;
76 g0    BitplaneAdr = HCBitmap->Planes[WhichPlane];
77 1f    Breite = HCScreen->Width;
78 YH    Hoehe = HCScreen->Height;
79 hc    Max = MaxPoints[GMode]/(ULONG)Breite;
80 nU    sprintf(Buffer,"\nDruckpunkte pro Bildschirmpunkt\nin der
         Breite (max.%d) ? ",Max);
81 xN    Write(HCWindow,Buffer,strlen(Buffer));
82 Wt    XFaktor = (UBYTE)ReadScr();
83 ov    WriteScr("Druckpunkte pro Bildschirmpunkt\nin der Hoehe ?
         ");
84 ay    YFaktor = (UBYTE)ReadScr();
85 5E    AnzSpalten = Breite*(USHORT)XFaktor;
86 VC    MemNeeded = AnzSpalten*(USHORT)VertBytes;
87 fo    InitGraphic[3] = (UBYTE)(AnzSpalten&255);
88 7R    InitGraphic[4] = (UBYTE)(AnzSpalten/256);
89 61    if ((Error = OpenDevice(PARALLELNAME,0,&IORpar,0)) != 0)
90 Or    {
91 9S6     printf("Can't open parallel.device, Error %d !\n",Error
         );
92 ig      CloseLibrary(IntBase);
93 1x      exit(99);
94 W13   }
95 1D    HCPort = CreatePort(PARALLELNAME,0);
96 ql    IORpar.IOPar.io_Message.mn_ReplyPort = HCPort;
97 So    Close(HCWindow);
98 AL    WritePrt(&GraphicAbstand[0],3);
99 pK    for (y=0;y<Hoehe;y++)
100 Y1   {
101 bg6    a = BitplaneAdr+(y*80);
102 yk     for (x=0;x<Breite;x++)
103 b4     {
104 LV9      z = (*(a+(x>>3)) & (1<<(7-((UBYTE)(x&7))))) !=0;
105 PN       r1 = (USHORT)(x*XFaktor);
106 14       for (h=0;h<XFaktor;h++)
107 sDC        PrtLine[r1+h] = z;
108 kF6    }
109 rB     for (h=0;h<YFaktor;h++)
110 rg9      Print();
111 nI3  }
112 2z   WritePrt(&InitGraphic[0],5);
113 LY   WritePrt(&PrtBuffer[0],MemNeeded);
114 Ia   WritePrt(&LineFeed,1);
115 1Z   WritePrt(&NormalAbstand[0],2);
116 Sd   CloseDevice(&IORpar);
117 75   CloseLibrary(IntBase);
118 pm   DeletePort(HCPort);
119 vQ0 }
120 tY  void WritePrt(Data,Length)
121 Ly  UBYTE *Data;
122 Ha  long Length;
123 v0  {
124 fE3   long Error;
125 n4    IORpar.IOPar.io_Data = (APTR)Data;
126 5r    IORpar.IOPar.io_Length = Length;
127 9j    IORpar.IOPar.io_Command = CMD_WRITE;
128 kI    if ((Error = DoIO(&IORpar)) != 0)
129 eR6     printf("parallel.device Write Error %d !\n",Error);
130 6b0 }
131 C3  void WriteScr(Data)
132 W9  UBYTE *Data;
133 5Y  {
134 bB3   sprintf(Buffer,Data);
135 pF    Write(HCWindow,Buffer,strlen(Buffer));
136 Ch0 }
137 xr  ULONG ReadScr()
138 Ad  {
139 KD3   ULONG Wert;
140 T3    Read(HCWindow,&Buffer,MAX_BUFFER-1);
141 eG    sscanf(Buffer,"%d",&Wert);
142 63    return (Wert);
143 Jo0 }
144 Sm  void Print()
145 Hk  {
146 ZB3   register USHORT x,y,z;
147 sx    y = AktBit;
148 PU    for (x=AktByte,z=0;z<AnzSpalten;x+=VertBytes,z++)
149 pM6     if (PrtLine[z] != 0)
150 HG9       PrtBuffer[x] += y;
151 I13   if ((AktBit >>= 1) == 0)
152 Or    {
153 6s6     AktBit = 128;
154 hA      if ((++AktByte) == VertBytes)
155 Ru      {
156 Cj9       AktByte = 0;
157 1i        WritePrt(&InitGraphic[0],5);
158 4H        WritePrt(&PrtBuffer[0],MemNeeded);
159 1J        WritePrt(&LineFeed,1);
160 aw        for (x=0;x<MemNeeded;x++)
161 sqC         PrtBuffer[x] = 0;
162 c76     }
163 d83   }
164 e90 }
(C) 1987 M&T
```

Listing. Der C-Quellcode von »Speedhardcopy« (Schluß)

PRINT TECHNIK

ÖSTERREICH · 1060 WIEN · STUMPERGASSE 34 · TEL. 0222/5973423 · TELEX 112996

SCHWEIZ · MICROTRON · 2542 PIETERLEIN · BAHNHOFSTR. 2 · TEL. 032/872429

ST-SCANNER HAWK CP 14 ST

Generalvertrieb BRD (auch für PC + Amiga)

Fakten:

Scannerelement: CCD-Sensor, 2048 Zeilen
Originale: Blätter und Objekte bis A4
Schnittstelle: Centronics Parallel
Betriebsarten: Scanner, 16 Graustufen, Kopierer und Thermoprinter, Telefax (Option)
Auflösung: 8 Punkte/mm, 200 dpi
Geschwindigkeit: 10 Sekunden für DIN A4, Hardcopy in 2 Sekunden, 500 Zeichen pro Sek.!!!
Kompression: Grafik bis Faktor 4, Dokument bis Faktor 20
Zoomfaktor: 0,1 bis 10,0
Kompatibel zu: Degas Elite, Stad, Word+, Profi Painter, Monostar, Fleet Street Publisher, Publishing Partner u.v.a.

Die Schriftenerkennung ist in Vorbereitung

SCANNER DM 2998,– inkl. Soft

Demodisk u. Unterlagen für DM 20,- anfordern!

VIDEO DIGITIZER PRO (1024x512) 8805 .... DM 498,-
VIDEO DIGITIZER REALTIZER PLUS .... DM 198,-
SPEICHERSCOPE .... DM 498,-
GENLOCK INTERFACE (extern steuerbar) ...... DM 1498,-

NEUHEITEN

1. VIDEO TEXT MODUL mit Soft .... DM 248,-
Schauen Sie zwischen den Zeilen Ihres Fernsehprogrammes nach dem Wetter/der Börse/dem Flugplan etc.

2. Clock-Modul für den ROM-Port mit Soft .... DM 98,-

Demodisk: DM 15,- Katalog anfordern! (DM 3,-) Täglich Versand

Computerperipherien

8000 München 40 · Nikolaistr. 2
Tel. 089/368197, Katalog DM 3,–
Tägl. Versand · Telex 523203d

Vesalia Versand

Soft- und Hardware Entwicklung · Produktion

NEU NEU NEU NEU NEU

EASYL — Zeichentablett für Amiga 500 — **678,- DM**

512 KB für A 500 mit Uhr, abschaltbar — **249,- DM**

2 MB für Amiga 500, abschaltbar — **865,- DM**

2 MB Golem RAM-Box für Amiga 1000 autokonfigurierend, Amiga-farbenes Metallgehäuse, Ein-/Aus-Schalter — **948,- DM**

VLW 2 3½"-Laufwerk (TEAC FD 35 FN) Amiga-farbenes Metallgehäuse, abschaltbar mit durchgeführtem Floppybus — **288,- DM**

VLW 4 3½"-Slimline-Laufwerk (NEC 1036 A) abschaltbar, mit durchgeführtem Floppybus, Amiga-farbenes Metallgehäuse, farblich passende Blende — **349,- DM**

VLW 7 3½"-Slimline-Laufwerk (NEC 1036 A) helle Blende, mit Zubehör u. Einbauanleitung, bereits **modifiziert** als internes Laufwerk für **Amiga 2000** — **268,- DM**

VLW 8 5¼"-Laufwerk (TEAC FD 55 FR) Amiga-farbenes Metallgehäuse, abschaltbar, farblich passende Blende, durchgeführter Bus, Amiga- u. MS-DOS-kompatibel, 40/80 Tr. — **449,- DM**

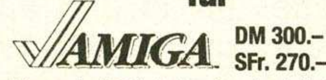
Alle Laufwerke mit 12 Monaten Garantie

Vesalia Versand
G. Does
4230 Wesel, Marienweg 40
Tel. 0281/65466 u. 62205
Ladenverkauf: Kornmarkt 23

Die schnellsten

Modula-2

Software-Entwicklungssysteme für



DM 300.– +MWSt.
SFr. 270.–

Extrem schneller Single-Pass-Compiler, in Workbench integriert, volle Unterstützung aller dokumentierter Amiga-Funktionen (Intuition, Exec, Grafik, usw.) Typen doppelter Genauigkeit und FFP, erzeugt schnellsten Maschinencode, linkt in wenigen Sekunden! Das komplette Entwicklungssystem umfasst Editor, Compiler, Linker, Module, deutsche Bedienungsanleitung und englisches Einführungsbuch in Modula-2.
Minimalkonfiguration: 512 kByte, 1 Laufwerk.

Zuschlag für zusätzliches deutsches Einführungsbuch DM 35.–/SFr. 30.–.
Demodiskette DM/SFr. 10.–

IBM/370-Mainframes — SFr. 16000.–

Einer der schnellsten Compiler der Welt (Single-Pass, 36000 Zeilen pro Minute), volle 32-Bit-Arithmetik, getrennte Übersetzung mit allen Vorteilen von Modula-2 (Versionskontrolle, Kompatibilitätsprüfung, Typechecking über die Modulgrenzen hinweg!), Schnittstellen zu Assembler und Fortran, Unterstützung von Projektbibliotheken, erzeugt schnellsten Native-Code (mit Arithmetik-Check) für Linker und Loader.
Jährlicher Wartungsvertrag SFr. 2750.–

IBM PC und Kompatible
DM 299.90 +MWSt./SFr. 267.50

Mit M2SDS entwickeln Sie Ihre Sokftware in einer komfortablen Fensterumgebung, welche alle Werkzeuge optimal integriert:
- Syntaxgesteuerter Editor
- inkrementeller Compiler, ist ein Vielfaches schneller als konventionelle Compiler
- schneller Linker, produziert direkt EXE-Programme
- Bibliotheksmanager, Module benötigen wenig Platz und sind übersichtlich geordnet
- Uhr, ASCII-Tabelle, Rechner
- alle Module im Sourcecode

M2SDS wird mit einem deutschen Handbuch geliefert, unterstützt den 8087-Prozessor, rechnet mit 18 Stellen Genauigkeit und bietet hervorragende Unterstützung des PC-DOS. Programme können bis 640 kByte lang sein. Zu keinem Software-Entwicklungssystem gibt es so viele Werkzeuge und Toolboxen wie für M2SDS.

M2SDS-Demodisketten — DM/SFr. 10.–

Turbo-Pascal nach Modula-2-Converter
DM 95.– +MWSt./SFr. 80.–

Bezugsquellen:

Bundesrepublik Deutschland:
- Interplan, Haslacher Weg 95, 7900 Ulm, 0731/2 69 32, 089/123 40 66
- SOS Software Service GmbH, Alter Postweg 101, 8900 Augsburg, 0821/85737
- SW-Datentechnik, Raiffeisenstr. 4, 2085 Quickborn, 04106/39 98
- Wilken & Sabelberg, Kasernenstr. 26, 3300 Braunschweig, 0531/34 71 21
- ALUDOM, Schlossstr. 62, 7000 Stuttgart, 0711/61 85 02/62 83 58

Schweiz:
- Frei-Elektronik, Stationsstr. 37, 8604 Volketswil, 01/945 54 32

Österreich:
- ICA GmbH, Heigerleinstr. 9, 1160 Wien, 0222/454 50 10

oder bei Ihrem nächsten Computer- oder Buchhändler

Generalvertrieb für Europa:
A. + L. Meier-Vogt
Im Späten 23
CH-8906 Bonstetten/ZH
Tel. (41) (1) 700 30 37

E-Mail: APLUSL@komsys.ifi.ethz.ch (UUCP)





DR. RUPRECHT

KOMMENTIERTES ROM-LISTING

Band 1: Betriebssystem-Dokumentation der Bereiche **EXEC, BOOT-ROM, DOS-BOOT.** Listing mit ausführlichen Kommentaren ca. 270 Seiten
Band 2: Dokumentation der **Resources** u. **Devices** ca. 300 Seiten — **je DM 69,–**



Astrologie
für Profis und Anfänger
in 'C' mit HR-Graphik

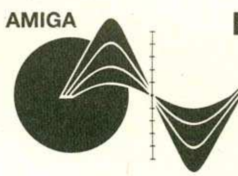
Programm des Lebens

● Schnelle Berechnung von **Geburts-, Composit-, Solarhoroskop, Transiten** und **Aspekten** ● Häuserberechnung nach **Koch** und **Placidus** ● **Karte** für Europa und U.S.A. zur Koordinatenbestimmung
Begleitbuch ca. 100 Seiten — **DM 149,–**

Erweiterungspaket (Atari ST)

Konstellations-Suche (Zeitbereich oder Diskette) **+ Text + Listengenerator** zur Erweiterung oder Änderung von Ausdruckformat, Textinhalten, Aspekten u. Koordinaten
2 Disketten mit Anleitung — **DM 175,–**

AMIGA



BIO-TIMER
Experimente mit dem Biorhythmus

● Darstellung von **2 Kurven** gleichzeitig ● Parameter für **Kurvenverschiebung** (Schwingungsdauer und Phasenlage) ● Einstellmöglichkeit **jahreszeitlicher Abhängigkeiten** ● **Berechnungsautomatik** ● **Mondphasen-Uhr** ● **Auswertung**
30 Seiten Begleitbuch — **DM 69,–**

Bestell-Coupon

Name: ....
Str.: ....
PLZ/Ort: ....
Programm: ....
Rechner: ....

Vorkasse (Scheck) ☐ Nachnahme (+DM 2,50) ☐



BIOSYSTEMS
SRI GmbH
Hansjakobstraße 122
8000 München 82

Auf zu den Sternen

Wer die Sterne am Himmel beobachtet, wird durch »Sternenhimmel« in der Suche nach bestimmten Sternen, Sternbildern und Planeten tatkräftig unterstützt. Für Profis und auch für Laien ist das Programm eine wertvolle Hilfe.

Sternenhimmel ist ein Programm, das die Fixsterne, die Planeten, Sonne und Mond für jeden beliebigen Punkt der Erde darstellt. Jedes Datum in unserem Jahrhundert sowie eine beliebige Uhrzeit kann eingestellt werden. Man kann Namen von sichtbaren Planeten, Sternen und Sternbildern erfragen oder bekannte Namen eingeben und danach suchen lassen. Die Wanderung der Planeten, Sonne und Mond kann über einen größeren Zeitraum beobachtet werden. Darüber hinaus zeigt »Sternenhimmel« die Mondphasen (Voll- oder Neumond und so weiter) an.

Das Programm (Listing) ist mit Pull-Down-Menüs ausgestattet, welche eine einfache und bequeme Bedienung ermöglichen.

Es ist reizvoll zuzuschauen, wie sich der Bildschirm (siehe Bild) aufbaut — für jeden erscheinenden Stern wird dessen Name und der Name des dazugehörenden Sternbildes angezeigt.

Nach dem Start von Sternenhimmel werden Sie aufgefordert, ein Datum in der Form »TT.MM.JJ« anzugeben. Fehleingaben werden natürlich zurückgewiesen.

Danach wird die Länge und Breite des gewünschten Standortes erfragt. Voreingestellt sind 50 Grad Nord und -7 Grad West (also 7 Grad Ost). Diese Angaben akzeptieren Sie mit einem Druck auf <RETURN>. Die Längen- und Breitengrade Ihres eigenen Standorts können Sie mit Hilfe eines Atlanten feststellen.

Jetzt müssen Sie noch die Uhrzeit (in Normalzeit) eingeben und die Voreinstellungen sind abgeschlossen.

Es erscheint eine dunkelblaue Kreisfläche, die den Himmel in Blickrichtung Süden darstellt. Der kleine Kreis ist der Himmelspol, nahe des Nordsterns (wenn Sie von der nördlichen Halbkugel der Erde beobachten).

Jetzt können Sie den Aufbau des »Himmels« mit einem Mausklick unterbrechen, beispielsweise um Einstellungen vorzunehmen oder falls Sie sich nur die Wanderung der Planeten ansehen wollen. Sie müssen also nicht warten, bis jeder Fixstern errechnet ist, wenn Sie diese noch gar nicht sehen wollen. Allerdings können Sie, wenn der »Himmel« noch nicht ganz aufgebaut ist, nicht nach Planeten, Sternen oder Sternbildern fragen.

Wenn Sie den Aufbau nicht unterbrechen, werden zuerst die Fixsterne dargestellt. Es folgen in dieser Reihenfolge die äußeren Planeten, die Sonne, die inneren Planeten und der Mond. Dies ist wichtig, sonst steht vielleicht einmal der Jupiter vor der Sonne und der Mond hinter dem Mars.

Die Fixsterne werden weiß, die Sonne gelb und die Planeten orange dargestellt. Der Mond wird auch weiß gezeichnet, er kann aber verschiedene Formen annehmen.

Jeweils drei bis vier Tage vor dem Ereignis nimmt der Mond die neue Form an. Er behält diese dann ungefähr sieben Tage lang. Auch in der Natur kann man beobachten, daß der Mond zirka drei Tage vor und auch ungefähr drei Tage nach Vollmond schon beziehungsweise noch ziemlich rund aussieht.

Links oben auf dem Bildschirm sehen Sie Datum, Normal- und Weltzeit. Rechts oben sind Länge und Breite dargestellt. Links unten wird während des Aufbaus der Name des Sterns oder Planeten und so weiter aufgeführt. Darunter erscheint der gebräuchliche Name des Sternbildes, dem der Stern angehört. Wiederum unter dieser Angabe finden Sie den lateinischen Namen des Sternbildes.

Nachdem der Bildschirm fertig aufgebaut ist, stehen folgende Menüs und Untermenüs zur Verfügung:

- Sternenhimmel

Neustart — Das Programm wird neu gestartet, alle eingestellten Werte sind verloren

Datum einstellen — Sie können ein neues Datum zur Himmelsbeobachtung einstellen

Uhrzeit einstellen — Uhrzeit neu einstellen

Länge und Breite — Sie können den Standort wechseln

für XX:XX neu zeichnen — in diesem Menüpunkt läuft die Uhr mit. Wenn Sie sich zum Beispiel bereits eine halbe Stunde mit einem Sternenhimmel beschäftigt haben, wählen Sie diesen Menüpunkt, um den »Bildschirmhimmel« zu aktualisieren.

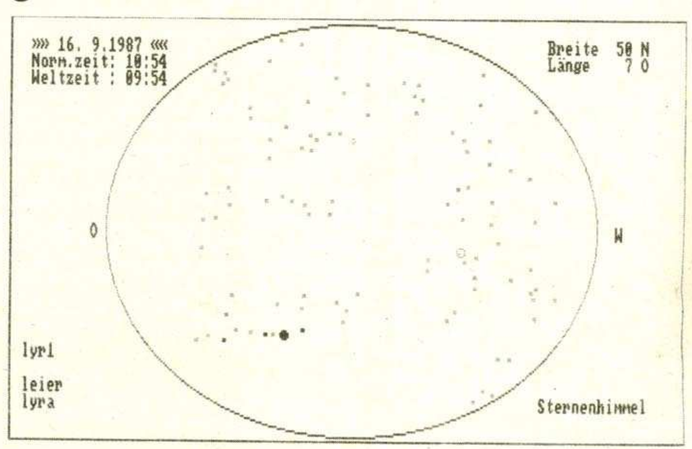


Bild. So erscheint der Himmel auf dem Bildschirm. Hier der Himmel am 16. September um 10.54 Uhr.

Ende — das Programm wird beendet. Manchmal wird der Screen nicht richtig geschlossen. Klicken Sie ihn einfach nach hinten und geben Sie »screen close 1« ein.

- Benennen

Klicken Sie in die Nähe eines Sterns oder eines Sternbildes. Dann wählen Sie zwischen:

Stern oder Planet — Der Pfeil wird genau auf den nächsten Stern oder Planet ausgerichtet. Links unten im Bild erscheint dessen Name.

Stern im Sternbild — Der Pfeil wird genau auf den nächsten Stern ausgerichtet. Das dazugehörige Sternbild blinkt. Der Name von Stern und Sternbild erscheint links unten im Bild.

Wenn Sie einen dieser beiden Menüpunkte anwählen, ohne einen Stern, Planet oder Sternbild angeklickt zu haben, werden Sie mit einer Fehlermeldung darauf aufmerksam gemacht. Nach Klicken mit dem linken Mausknopf verschwindet diese Meldung.

- Zeigen

Planet, Stern oder Sternbild — geben Sie den Namen des Planeten, Sterns oder Sternbildes, das Sie suchen, in Kleinbuchstaben ein. Der erste Teil des Namens genügt. Nach kurzer Suche wird, wenn sichtbar, ein Pfeil auf den gesuchten Planeten oder Stern gerichtet. Sternbilder blinken. Links wird der volle Name des Objektes aufgeführt.

Planetenwanderung — Sie können angeben, wie viele Schritte Sie beobachten wollen. Danach wird festgelegt, wieviel Zeit zwischen diesen Schritten vergehen soll. Beispiel: Sie möchten die Planeten über 30 Tage hin beobachten. Für die gleiche Stunde eines jeden Tages sollen die Planeten gezeichnet werden.

Demnach geben Sie 30 Schritte und 24 Stunden ein. Wenn die Schritte abgearbeitet sind, wird Sternenhimmel wieder für die eingestellte Zeit gezeichnet.

Sonnen- und Mondwanderung — Sie beobachten den Verlauf von Sonne und Mond

Planeten, Sonne und Mond — Bewegungen von Planeten, Sonne und Mond beobachten

- Hilfe

In diesem Menü finden Sie einige Erläuterungen zu »Sternenhimmel«

Achten Sie beim Programmstart darauf, daß alle unnötigen Fenster geschlossen werden. Dies ist wichtig, da »Sternenhimmel« zu den speicherintensiveren Programmen gehört.

(Arno Gölzer/rb)

Programmname:	Sternenhimmel
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2

```
Programm : Sternenhimmel
------------------------

 1 FnO CLEAR,50000&
 2 wU  ON ERROR GOTO erro
 3 mV  RESTORE pfeil
 4 oX  FOR i=1 TO 84
 5 cB1  READ a
 6 hJ   pf$=pf$+CHR$(a)
 7 CHO NEXT
 8 MF  SCREEN 1,640,220,3,2
 9 PY  WINDOW 2,,(0,0)-(631,186),16,1
10 Zv  WINDOW OUTPUT 1
11 Le  FOR i=0 TO 3
12 MK1  PALETTE i,0,0,0
13 INO NEXT
14 j1  WINDOW OUTPUT 2
15 5x  FOR i=0 TO 1 STEP .003
16 r41  PALETTE 0,i,i,i
17 MRO NEXT
18 hR  PALETTE 1,.5,.2,.3
19 wd  PALETTE 2,.2,.3,.6
20 aR  PALETTE 4,1,1,.13
21 S6  PALETTE 5,.7,.5,.3
22 K7  PALETTE 6,.9,.3,.1
23 Jt  PALETTE 7,.8,.8,.8
24 tB  WINDOW OUTPUT 2
25 3x  MENU 1,0,1,"Sternenhimmel"
26 j1  MENU 1,1,1,"Neustart"
27 9t  MENU 1,2,1,"Datum  einstellen"
28 sr  MENU 1,3,1,"Uhrzeit einstellen"
29 61  MENU 1,4,1,"L"+CHR$(228)+"nge und Breite"
30 gl  MENU 1,5,1,""
31 GA  MENU 1,6,1,"E N D E "
32 oA  MENU 2,0,1,"Benennen"
33 XQ  MENU 3,0,1,"Zeigen"
34 FD  MENU 3,2,1,"Planetenwanderung"
35 AW  MENU 3,3,1,"Sonnen- und Mondwanderung"
36 iY  MENU 3,4,1,"Planeten, Sonne und Mond"
37 zC  MENU 4,0,1,"Hilfe"
38 WR  MENU 4,1,1,"Erkl"+CHR$(228)+"rung"
39 XC  MENU 4,2,1,"Mondphasen"
40 M3  MENU 4,3,1,+CHR$(169)+"opyright"
41 Vd  ON MENU GOSUB auswertung
42 tp  ON MOUSE GOSUB maus
43 N5  GOTO beginn
44 2M  maus:
45 ax  pw=0:pwa2=0:sm=0:pf=0:npo=0
46 T1  IF frei THEN ok=1:OBJECT.OFF:GOTO lo
47 20  such=1
48 ot  of=MOUSE(0)
49 5X  xx=MOUSE(1)
50 Hh  yy=MOUSE(2)
51 JP  OBJECT.SHAPE 2,pf$
52 et  OBJECT.X 2,xx+1
53 q8  OBJECT.Y 2,yy+1
54 D6  OBJECT.ON
55 OR  GOTO loop
56 MD  auswertung:
57 D7  MENU OFF
58 8s  m1=MENU(1)
59 pA  ON MENU(0) GOTO a,b,c,d
60 EN  beginn:
61 OK  LINE (2,2)-(628,60),2,bf
62 BG  TIMER ON:COLOR 3,2
63 PF  LOCATE 4,30:PRINT "STERNENHIMMEL"
64 FT  LINE (226,32)-(340,32),3
65 EN  RANDOMIZE TIMER
66 Jt  FOR i=1 TO 30
67 151  zuf:
68 Ky   x=RND*627:y=RND*59
69 2H   IF x<2 OR y<2 THEN zuf
70 EJ   PSET (x,y),0
71 k90 NEXT:
72 vD  COLOR 1,0
```

Listing. »Sternenhimmel« zeigt Ihnen die Position aller Himmelskörper auf dem Bildschirm an. Das Listing bitte mit dem Checksummer (siehe Seite 76) eingeben.

```
73 Im   IF nda THEN RETURN
74 CJ   IF nlb THEN breite
75 7R   datum:
76 Pz   TIMER OFF
77 RG   LOCATE 10,1:PRINT "Bitte Datum eingeben (TT.MM.JJ):
78 J4   LOCATE 10,34:INPUT "",da$:IF LEN(da$)<8 THEN datum
79 Zu   IF MID$(da$,3,1)<>MID$(da$,6,1) OR MID$(da$,6,1)<>"."THEN da
        tum
80 Mt   j$="19"+RIGHT$(da$,2):j=VAL(j$)
81 Zu   t$=LEFT$(da$,2):t=VAL(t$):IF t<1 OR t>31 THEN PRINT:PRINT "T
        ag falsch eingegeben ! ("t")           ":GOTO datum
82 fq   m$=MID$(da$,4,2):m=VAL(m$):IF m<1 OR m>12 THEN PRINT:PRINT "
        Monat falsch eingegeben ! ("m")          ":GOTO datum
83 At   ch=0:GOSUB check:IF ch=1 THEN PRINT :PRINT "Tag falsch einge
        geben ("t$"."m$")":GOTO datum
84 MN   a$=t$+"."+m$+"."+j$:PRINT:PRINT a$"
        "
85 YY   IF ndat THEN: FOR n1i=1 TO 2000:NEXT: RETURN
86 2P   i1=124:i2=125:i3=22:F=2:such=0:ok=0:pf=0:erla=0
87 Zn   pi=3.1416:p1=pi/180:p2=180/pi:c=360
88 Ae   DIM z$(i3,1),z(i3,1),p(i1),z%(2*i2),rel(i1),del(i1),nn$(i1),d
        1(i1),moa(12)
89 uF   RESTORE mondat
90 CY   FOR ii=1 TO 12
91 uj1   READ moa(ii)
92 Ze0  NEXT
93 YL   breite:
94 JP   LOCATE 15,1:PRINT "Breite   (50) :                    "
95 Ib   LOCATE 15,17:INPUT "",b$:IF b$="" THEN b$="50"
96 vQ   b=VAL(b$)
97 OF   IF ABS(b)>=90 THEN breite
98 ft   LOCATE 15,16:PRINT b
99 It   laenge:
100 v6  LOCATE 16,1:PRINT "L"CHR$(228)"nge    (-7) :
        "
101 vL  LOCATE 16,17:INPUT "",l$:IF l$="" THEN l$="-7"
102 pe  l=VAL(l$)
103 5T  IF ABS(l)>=180 THEN laenge
104 ud  LOCATE 16,16:PRINT l
105 DA  sb=SIN(b*p1):cb=COS(b*p1)
106 dc  IF nlb THEN RETURN
107 H5  uhrzeit:
108 Vy  LOCATE 19,1:PRINT "Uhrzeit (HHMM):                    "
109 Sr  LOCATE 19,17:INPUT "",t1$
110 yr  IF VAL(RIGHT$(t1$,2))>59 THEN uhrzeit
111 uM  IF LEN(t1$)<>4 THEN uhrzeit
112 lo  IF VAL(t1$)>2400 OR VAL(t1$)<0 THEN uhrzeit
113 Un  pwand1:
114 9M  wz=INT(VAL(t1$)-100)
115 c   IF wz<100 THEN wz=wz+2400
116 KF  wz$=STR$(wz)
117 Us  wz=VAL(wz$)/100
118 jG  nspr:
119 NB  jz=j-1900:moa=moa(m):IF (jz/4=INT(jz/4) OR jz=0) AND m>2 THEN
        moa=moa+1
120 9S  mond=(INT(jz/4)+1+(jz*365)+moa+t)/29.44889
121 X1  mond=(mond-INT(mond))*10:IF pw THEN ntimer
122 RI  TIMER ON:ON TIMER(60) GOSUB zeit
123 40  ntimer:
124 nP  zt=INT(wz)+(wz/.6-INT(wz/.6))
125 qx  i=m<3
126 Oz  k=t+INT((153*m-11*i-162)/5)+INT((1461*j+i)/4)+(j>=0)*366
127 NR  IF k>577736& THEN k=k-INT((INT((j+1)/100)*3-5)/4)
128 AM  ta=k-693596&:tg=k-711858&+zt/24:i=ta/36525&
129 ua  e=23.452294#-i*.013125-i*i*1.639E-06+i*i*i*5.028E-07
130 Xb  es=SIN(e*p1):ec=COS(e*p1)
131 1B  IF pw THEN pwand2
132 TT  REM funktionen
133 Ls  DEF FN ass(x)=ATN(x/SQR(1-x*x))
134 5H  DEF FN ac(x)=pi/2-ATN(x/SQR(1-x*x))
135 11  DEF FN mo(x)=x-INT(x/c)*c
136 OM  REM zeit
137 uE  pwand2:
138 8   wz$=MID$(STR$(INT((wz-INT(wz))*100+.5)),2)
139 F1  wz$=RIGHT$("0"+wz$,2)
140 vJ  wz$=RIGHT$(" "+STR$(INT(wz)),3)+":"+wz$
141 Åj  lo=INT(l/15)*15
142 3E  oz=FNmo(zt*15-lo)/15
143 XL  oz$=RIGHT$(" "+STR$(INT(oz)),3)+RIGHT$(wz$,3)
144 F8  REM aries
145 qc  ar=zt*360.985647#/24+((ta/1461)-INT(ta/1461))*1440.02509#
146 hx  ar=ar+INT(ta/1461)*.0307572+99.2018973#
147 79  ar=FNmo(ar)
148 vM  IF pw THEN RETURN
149 Ue  RESTORE sternbilder
150 ee  FOR a=0 TO i3
151 b81  READ z$(a,0),z$(a,1),z(a,0),z(a,1)
152 XcO NEXT
153 rm  bschirm:
154 BM  MOUSE OFF:CLS
155 oY  nar$="Moment...":GOSUB wtas:nar$="Klick ...":WINDOW OUTPUT 2
156 UW  MENU 2,1,1,""
157 Zc  MENU 2,2,1,""
158 Yb  MENU 3,1,1,""
159 Ww  GOSUB bildschirm:MOUSE ON:IF bs THEN RETURN
160 li  GOSUB erde
161 WM  GOSUB fixsterne
162 bq  GOSUB planeten
163 AT  MENU 2,1,1,"Stern oder Planet "
164 dX  MENU 2,2,1,"Stern im Sternbild"
165 un  MENU 3,1,1,"Planet, Stern oder Sternbild"
166 1P  loop:
167 xl  bs=0:frei=0:COLOR 1,0
168 7o  MOUSE ON:MENU ON
169 Oh  WINDOW CLOSE 3
170 Tp  WINDOW CLOSE 4:COLOR 3,0
171 Dn  LOCATE 23,59:PRINT "     Sternenhimmel     ":COLOR 1,0
172 JW  lo2:
173 sY  g$=INKEY$:IF g$=""THEN lo2
174 7V  loop1:
175 7S  p=ASC(g$)
176 FZ  loop2:
177 Ot  IF p=49 THEN LOCATE 23,59:PRINT "sucht Stern            ":GOTO st
        ern
178 oQ  IF p=50 THEN LOCATE 23,59:PRINT "sucht Sternbild        ":GOTO st
        ern
179 b9  IF p=51 THEN start
180 OD  IF p=52 THEN ende
181 mJ  IF p>64 AND p< 91 THEN lesen
182 3p  GOTO lo2
183 Jm  a:
184 X4  ON m1 GOTO start,datn,uhr,lub,nz,ende
185 Nr  b:
186 Q3  ON m1 GOTO ste,stb
187 Rw  c:
188 9V  ON m1 GOTO lesen,pwanderung,sm,psm
189 V1  d:
190 vX  ON m1 GOTO erklaerung,mondphasen,rechte
191 bf  start:
192 Fe  RUN
193 fn  nz:
194 Hi  erla=0:GOTO pwand1
195 in  datn:
196 zb  CLS : erla=0
197 97  dat=1:LOCATE 15,1:PRINT "Breite :"b
198 kL  LOCATE 16,1:PRINT "L"CHR$(228)"nge  :"l
199 eh  LOCATE 18,1:PRINT "Zeit    :"h1$":"m1$:GOSUB beginn:ndat=0
200 DP  GOTO nspr
201 AF  lub:
202 cZ  CLS : erla=0 : nlb=1
203 F1  LOCATE 12,1:PRINT "Datum  :"a$
204 HY  LOCATE 18,1:PRINT "Zeit   :"h1$":"m1$
205 dN  GOSUB beginn:nlb=0:GOTO nspr
206 wD  uhr:
207 Am  CLS : erla=0
208 RW  nda=1:GOSUB beginn:nda=0
209 L7  LOCATE 12,1:PRINT "Datum  :"a$
210 07  LOCATE 15,1:PRINT "Breite :"b
211 xY  LOCATE 16,1:PRINT "L"CHR$(228)"nge  :"l
212 yL  GOTO uhrzeit
213 8A  pwand:
214 U3  bs=1:frei=1MOUSE OFF:TIMER OFF:WINDOW 3,,(215,70)-(415,112),
        0,1:WINDOW OUTPUT 3
215 F1  PRINT wa$"- Wanderung":PRINT:INPUT " Wieviele Schritte";n$:sc
        hritte=VAL(n$)
216 i4  PRINT " Zeitabst"+CHR$(228)+"nde":INPUT" in Std. (max=24) ";n
        $:ezeit=INT(VAL(n$)):IF ezeit >24 THEN pwand
217 Ft  wzu:
218 nU  WINDOW CLOSE 3
219 YZ  tret$=t1$:d1$=t$:d2$=m$:d3$=j$:n$="":n1$="":n2$="":CLS
220 hC  GOSUB bschirm:MOUSE ON
221 U1  FOR pwa=1 TO schritte
222 fq2   pw=1:t1=VAL(t1$)+ezeit*100:t1$=RIGHT$("0000"+STR$(t1),4)
223 lo    IF t1>2400 THEN t1=t1-2400:t1$=RIGHT$("0000"+STR$(t1),4):ch
          2=1:erla=0:GOSUB zeit:ch2=0:GOTO nzeit
224 U2    erla=0:GOSUB zeit
225 oo0 zeit:
226 Oh2   GOSUB erde:t=VAL(t$):j=VAL(j$):m=VAL(m$):erla=0:GOSUB pwand
          1:IF sm THEN smo
227 b8    GOSUB planeten:FOR ii=1 TO 200:NEXT ii
228 gC    RESTORE planet:pf=2:GOSUB pwand3:GOTO npw
229 IZ    smo:
230 nX    pf=4:GOSUB sonne:GOSUB sonne2:GOSUB mon:FOR ii=1 TO 500:NEX
          T ii
231 Vs    pf=2:GOSUB sonne:GOSUB sonne2:GOSUB mon
232 qD0 npw:
233 pe  NEXT pwa
234 JP  TIMER ON:bs=0:pwa2=0:pw=0:sm=0:t$=d1$:m$=d2$:j$=d3$:t1$=tret$
```

```
    :FOR ii=1 TO 2000:NEXT:GOTO pwand1
235 Sk  psm:
236 ZL  wa$=" P,S & M ":pwa2=1:GOTO pwand
237 uu  sm:
238 Nr  wa$=" S & M ":sm=1:GOTO pwand
239 Ou  pwanderung:
240 7v  wa$=" Planeten ":GOTO pwand
241 MG  ende:
242 C6  MENU OFF
243 Kg  WINDOW OUTPUT 1
244 8o  WINDOW CLOSE 2
245 ou  SCREEN CLOSE 2
246 R6  PALETTE 0,.2,.3,.6
247 pr  PALETTE 1,1,1,1
248 wJ  PALETTE 3,.8,.5,0!
249 1g  END
250 de  check:
251 sc  mon=VAL(m$):tag=VAL(t$):jahr=VAL(j$)
252 we  IF tag>31 THEN fe
253 nE  IF mon>7 THEN n3
254 Nw  IF mon=2 AND jahr/4=INT(jahr/4) AND tag>29 THEN fe
255 xc  IF mon=2 AND jahr/4<>INT(jahr/4) AND tag>28 THEN fe
256 oy  FOR ii=1 TO 7
257 ui1 IF ii<>mon THEN n1258 u5  IF ii/2=INT(ii/2) AND tag>30 THEN
    ch=1:ii=7
259 C90 n1:
260 HM  NEXT
261 7B  IF ch THEN fe
262 aC  RETURN
263 ML  n3:
264 Sv  FOR ii=8 TO 12
265 6v1  IF ii<>mon THEN n2
266 57   IF ii/2<>INT(ii/2) AND tag>30 THEN ch=1:ii=12
267 NLO n2:
268 PU  NEXT
269 FJ  IF ch THEN fe
270 1K  RETURN
271 eJ  fe:
272 94  ch=1:RETURN
273 hv  ste:
274 TA  p=49:GOTO loop2
275 X1  stb:
276 qP  p=50:GOTO loop2
277 fB  stern:
278 Ak  IF such=0 THEN WINDOW 4,,(217,77)-(417,102),0,1:PRINT:PRINT
    " Planet, Stern oder":PRINT " Sternbild anklicken":GOTO schr
279 k5  such=0
280 tC  OBJECT.OFF
281 cY  x=xx/2-1.5:y=yy-3
282 BD  xm=200000&
283 6H  IF p=50 THEN sternbild
284 9x  FOR a=0 TO 10 STEP 2
285 cl1  F=p(a)-x:h=p(a+1)-y
286 nX   xx=F*F+h*h
287 CP   IF xx<xm THEN xm=xx:zp=a
288 5A   IF xx=0 THEN a=12
289 kp0 NEXT
290 9b  n$=p$(zp/2):n1$=" ":n2$=" "
291 q6  IF xx=0 THEN weiter
292 dc  sternbild:
293 rU  xp=xm
294 qo  FOR a=0 TO i1
295 S61  F=z%(a)-x:h=z%(a+i2)-y
296 xh   xx=F*F+h*h
297 yt   IF xx<xm THEN xm=xx:z=a
298 32   IF xx=0 THEN a=i2
299 uz0 NEXT
300 YL  x=p(zp):y=p(zp+1)
301 CL  IF xp=xm THEN weiter
302 t0  x=z%(z):y=z%(z+i2)
303 1u  re=re1(z):de=de1(z):n$=nn$(z):d=d1(z)
304 Op  n1$=z$(d,1):n2$=z$(d,0)
305 TV  IF p=50 THEN z1=z:z=d:LOCATE 23,59:PRINT "Sternbild ...     "
    :GOTO PR
306 9U  weiter:
307 JW  GOSUB pri:GOSUB move:GOTO schr
308 4F  lesen:
309 S0  WINDOW 3,,(215,70)-(415,112),0,1
310 bp  WINDOW OUTPUT 3
311 pr  PRINT:PRINT "      Bitte den Namen"
312 eu  PRINT " (oder den  ersten Teil)"
313 mi  PRINT " des Gesuchten eingeben.":PRINT
314 NV  INPUT" ",n$WWINDOW CLOSE 3
315 Z9  p=0:z=-1
316 K1  FOR s=0 TO 5
317 vz1  IF n$=LEFT$(p$(s),LEN(n$)) THEN z=s:s=5
318 DI0 NEXT
319 r0  IF z<0 THEN pr1
320 v5  x=p(2*z):y=p(2*z+1)
321 hh  n1$=p$(z):n2$=" "
322 YJ  IF x=0 THEN fehler
323 V9  n$="":GOSUB move: GOTO schr
324 yH  pr1:
325 D3  FOR s=0 TO i3
326 x71  IF n$=LEFT$(z$(s,0),LEN(n$)) THEN z=s:s=i3
327 2D   IF n$=LEFT$(z$(s,1),LEN(n$)) THEN z=s:s=i3
328 NS0 NEXT
329 cB  IF z<0 THEN such
330 cv  z1=-1
331 zz  PR:
332 K2  a1=z(z,1):a2=a1+z(z,0)
333 hF  IF z1>=0 THEN ne
334 mI  FOR a=a1 TO a2
335 rE1  IF z%(a) THEN z1=a:a=a2
336 Va0 NEXT
337 EJ  IF z1<0 THEN n1$=z$(z,1):GOTO fehler
338 zm  ne:
339 DZ  F=1:ok=0:n1$=z$(z,1):n2$=z$(z,0)
340 YM  x=z%(z1):y=z%(z1+i2):GOSUB wtas:WINDOW OUTPUT 2:GOSUB pri
341 SN  lo:
342 SA  IF bs THEN loop
343 r1  IF ok THEN F=0
344 wS  FOR a=a1 TO a2
345 KQ1  x=z%(a)
346 4R   IF x THEN PSET (2*x,z%(a+i2)),F
347 g10 NEXT
348 pG  IF ok THEN loop
349 xM  F=F+1:IF F>3 THEN F=0
350 Jt  g$=INKEY$:IF g$<>"" THEN OBJECT.OFF:F=1:ok=1
351 Ft  FOR a=1 TO 65:NEXT:GOTO lo
```

Listing. »Sternenhimmel« zeigt Ihnen die Position aller Himmelskörper auf dem Bildschirm an. (Fortsetzung)

```
352 70  such:
353 WU  RESTORE fixst
354 bQ  FOR s=1 TO i1
355 aO1  IF n$=LEFT$(nn$(s),LEN(n$)) THEN z=s:s=i2
356 puO NEXT
357 wM  IF z<0 THEN f1
358 97  a=d1(z)
359 7a  IF z%(z)=0 THEN n1$=nn$(z):GOTO fehler :REM z%(z) war x
360 pK  x=z%(z):y=z%(z+i2)
361 QW  n$=nn$(z):n1$=z$(a,1):n2$=z$(a,0)
362 FT  GOSUB move:GOTO schr
363 oJ  fehler:
364 ss  n1$=LEFT$(n1$,15)
365 1k  WINDOW 4,,(217,77)-(417,102),0,1
366 8C  PRINT :PRINT" " n1$
367 9H  PRINT " nicht sichtbar !":GOTO schr
368 hW  f1:
369 EF  n$=LEFT$(n$,15)
370 vN  WINDOW 4,,(210,80)-(410,105),0,1
371 xu  PRINT :PRINT " "n$
372 hq  PRINT " nicht gespeichert !":GOTO schr
373 xI  move:
374 c8  OBJECT.SHAPE 2,pf$:
375 gT  OBJECT.X 2,2*x+3
376 1H  OBJECT.Y 2,y+3
377 QJ  OBJECT.ON
378 2o  GOTO pri
379 Sr  bildschirm:
380 YG  t2$=MID$(STR$(INT(b+.5)),2,4)
381 dA  sp$=RIGHT$("   ",3-LEN(t2$))
382 NZ  GOSUB zeit
383 v7  LOCATE 2,64
384 pv  PRINT "Breite "sp$;t2$;
385 On  b$="N":IF b<0 THEN b$="S"
386 Uz  PRINT" "b$
387 TL  t2$=MID$(STR$(INT(l+.5)),2,4)
388 kH  sp$=RIGHT$("   ",3-LEN(t2$))
389 5I  LOCATE 3,64
390 y3  PRINT "L"CHR$(228)"nge "sp$;t2$;
391 G5  l$="W":IF l<0 THEN l$="O"
392 O3  PRINT" "l$
393 OP  LOCATE 13,10:PRINT "O"
394 pc  LOCATE 13,72:PRINT "W":COLOR 2,0
395 MD  f1=1:f2=1:f3=1:PALETTE 2,f1,f2,f3
396 Vn  CIRCLE (320,98),230,,,,.42:CIRCLE (320,98),231,,,,.42
397 Qb  PAINT (320,98)
398 Ul  FOR ii=0 TO 100
399 9p2   f1=f1-.007:f2=f2-.008:f3=f3-.004:PALETTE 2,f1,f2,f3
400 XcO NEXT
401 uC  COLOR 1,0:hu=0:IF b<0 THEN hu=200
402 32  CIRCLE(320,hu+b/9*10),2,0,,,.5
403 rT  RETURN
404 B9  erde:
405 fo  pf=0
406 f4  smw:
407 w9  REM erde
408 54  be=FNmo(tg*.985609121#+99.18)
409 ZI  el=FNmo(be+SIN((be-102.2)*p1)*1.845)
410 1U  ea=1+SIN((el-192.2)*p1)*.0167
411 zb  RETURN
412 z5  sonne:
413 9C  ls=FNmo(el+180)
414 FQ  lm=tg*13.1763976#+51.23
415 ZH  pm=tg*.111399014#+208.9
416 aw  km=372.1-tg*.052953643#
417 zc  lm=FNmo(lm)
418 SD  pm=FNmo(pm)
419 uV  km=FNmo(km)
420 J4  am=lm-pm
421 SK  km=km-SIN(ass*p1)*.16
422 Uw  ms=(lm-ls)*2-am
423 C8  am=am+SIN(ms*p1)*1.27388889#-SIN(ass*p1)*(.18638889#+.36)
424 nc  lm=lm+SIN(ms*p1)*1.27388889#-SIN(ass*p1)*.18638889#+SIN(am*p1
        )*6.28833333#
425 45  m1=lm-ls
426 Cl  lm=lm+SIN(m1*2*p1)*.658333333#
427 1d  m2=lm-km
428 XW  lm=lm-SIN(m2*2*p1)*.12
429 iH  bm=SIN(m2*p1)*5.14539
430 mC  m3=(lm-ls)*2-m2
431 dk  bm=bm+SIN(m3*p1)*.15
432 Kw  RETURN
433 Ki  sonne2:
434 t8  n$="Sonne":n1$="                    ":n2$=n1$:al=ls:ab=0:npo=1:GOS
        UB rek
435 U2  IF x=0 THEN weok
436 D1  IF pf=3 THEN npf=4 ELSE npf=pf
437 TZ  FOR ii=0 TO 4
438 yJ  CIRCLE (2*x,y),ii,npf,,,.5
439 e2  NEXT:GOTO weok
440 Xj  mon:
441 xn  n$="Mond":n1$="                    ":n2$=n1$:al=lm:ab=bm:npo=1:GOS
        UB rek
442 b9  IF x=0 THEN weok
443 1Q  IF pf=3 OR pf=4 THEN npf=0 ELSE npf=pf
444 xC  IF mond <3.5 THEN CIRCLE (2*x,y),4,npf,4.8,1.6,.5:PSET (2*x+1
        ,y-1),npf:PSET (2*x+1,y),npf:PSET (2*x+1,y+1),npf:GOTO monok
445 5H  IF mond <6.5 THEN: FOR ii=4 TO 0 STEP -1:CIRCLE (2*x,y),ii,np
        f,,,.5:NEXT:GOTO monok
446 ke  IF mond <8.3 THEN CIRCLE (2*x,y),4,npf,1.6,4.8,.5:PSET (2*x-1
        ,y-1),npf:PSET (2*x-1,y),npf:PSET (2*x-1,y+1),npf:GOTO monok
447 Ck  CIRCLE (2*x,y),4,npf,,,.5
448 2f  monok:
449 3N  weok:
450 RU  npo=0:RETURN
451 sO  planeten:
452 Ka  RESTORE planet:npo=0:pf=3
453 2N  pwand3:
454 OI  FOR zz=0 TO 5
455 Ih1  READ n$,tb,ep,ph,mp,e,kn,i,ae
456 fJ   p$(zz)=n$
457 Sb   ml=FNmo(tb*tg+ep)
458 d6   wl=ml+SIN((ml-ph)*p1)*mp
459 2X   sp=ae+SIN((wl-ph-90)*p1)*e*ae
460 4N   ws=FNmo(c+el-wl)*p1:si=SIN(ws)
461 2V   fl=ea/sp-COS(ws)
462 Nk   we=ATN(si/fl)*p2
463 vO   al=FNmo(el+we-180*(fl>=0))
464 hw   wt=SIN((wl-kn)*p1)*i
465 XM   ab=ATN(TAN(wt*p1)*ABS(SIN(we*p1)/si))*p2
466 Lv   n1$="                    ":n2$=n1$:GOSUB rek
467 a8   p(2*zz)=x:p(2*zz+1)=y
468 6h   IF (pw=0 OR pwa2) AND zz=1 THEN GOSUB sonne:GOSUB sonne2
469 ejO NEXT
470 tt  IF pw=0 OR pwa2 THEN GOSUB sonne:GOSUB mon
471 bg  pf=0:RETURN
472 Dz  fixsterne:
473 SQ  RESTORE fixst
474 XM  FOR s=1 TO i1
475 hO1  READ re,de,n$,a
476 wa   re1(s)=re:de1(s)=de:nn$(s)=n$:d1(s)=a
477 Yb   de=de*p1
478 KI   n1$=z$(a,1):n2$=z$(a,0)
479 U2   GOSUB ss
480 dC   z%(s)=x:z%(s+i2)=y
481 qvO  NEXT
482 8k  RETURN
483 im  rek:
484 OH  sn=SIN(ab*p1):cs=COS(ab*p1)
485 Xb  sl=SIN(al*p1):cl=COS(al*p1)
486 CG  de=FNass(ec*sn+es*cs*sl)
487 Fp  re=2*ATN((ec*cs*sl-es*sn)/(COS(de)+cs*cl))
488 RO  re=FNmo(re*p2)
489 GM  ss:
490 i4  sw=FNmo(ar-re-l)
491 5k  sw=sw*p1
492 I1  h=FNass(sb*SIN(de)+cb*COS(de)*COS(sw))
493 Xd  IF h<0 THEN x=0:y=0:RETURN
494 X2  a=(SIN(de)-sb*SIN(h))/(cb*COS(h))
495 8U  IF ABS(a)>=1 THEN a=a*.9998
496 QV  a=FNac(a)
497 uz  IF sw<pi THEN a=-a
498 an  r=99-h*198/pi:w=-a-pi/2
499 Et  x=INT(COS(w)*r*1.1+159.5)
500 dX  y=INT(SIN(w)*r+99.5)
501 Jh  IF npo THEN pri
502 T4  PSET(2*x,y),pf
503 Tg  pri:
504 bf  IF pf=2 THEN RETURN
505 VH  n$=LEFT$(n$+"               ",13):sp$="                    "
506 E2  LOCATE 20,2:PRINT n$
507 aM  LOCATE 22,2:PRINT n1$+LEFT$(sp$,LEN(sp$)-LEN(n1$)):LOCATE 23,
        2:PRINT n2$+LEFT$(sp$,LEN(sp$)-LEN(n1$))
508 YA  RETURN
509 Ss  zeit:
510 9J  WINDOW OUTPUT 2 :ch=0 :fe=0
511 O8  h1=VAL(LEFT$(t1$,2))
512 4E  m1=VAL(RIGHT$(t1$,2))
513 1l  jj=VAL(j$):mm=VAL(m$):tt=VAL(t$):IF ch2 THEN chdat
514 Iy  IF erla=0 THEN erla=1:GOTO er
515 K2  m1=m1+1 :IF m1<60 THEN er
516 YU  m1=0:h1=h1+1:IF h1<24 THEN er
517 1v  IF h1=24 AND m1=0 THEN er
518 fp  h1=0
519 cO  chdat:
520 Xd  tt=tt+1:t$=STR$(tt):GOSUB check:IF ch=0 THEN er
521 Rw  tt=1:mm=mm+1:IF mm<13 THEN er
```

```
522 Uq  mm=1:jj=jj+1:IF jj<1999 THEN er
523 HG  fe=1
524 KB  er:
525 hz  j$=RIGHT$(STR$(jj),4):m$=RIGHT$(STR$(mm),2):t$=RIGHT$(STR$(tt
        ),2):a$=t$+"."+m$+"."+j$
526 12  IF fe THEN a$="Ung"+CHR$(252)+"ltig !"
527 XT  h1$=RIGHT$("00"+RIGHT$(STR$(h1),LEN(STR$(h1))-1),2):m1$=RIGHT
        $("00"+RIGHT$(STR$(m1),LEN(STR$(m1))-1),2)
528 vT  t1$=h1$+m1$:hwz=VAL(h1$)-1:IF hwz<0 THEN hwz=23
529 Lb  hwz$=RIGHT$("00"+RIGHT$(STR$(hwz),LEN(STR$(hwz))-1),2)
530 b0  MENU 1,5,1,"f"+HHR$(252)+"r "+h1$+":"+m1$+" zeichnen"
531 y0  LOCATE 2,3:PRINT CHR$(187)CHR((187)" "a$" "CHR$(171)CHR$(171)
532 yV  LOCATE 3,3:PRINT"Norm.zeit: " h1$":"m1$:PRINT "  Weltzeit : "
        hwz$":"m1$
533 xZ  RETURN
534 JX  schr:
535 vF  GOSUB wtas
536 BH  warte:
537 gi  IF INKEY$="" THEN warte
538 Ry  OBJECT.OFF:GOTO loop
539 yR  wtas:
540 dm  frei=1
541 gz  WINDOW 3,,(5,85)-(80,100),0,1
542 LZ  WINDOW OUTPUT 3
543 cw  COLOR 3,0
544 IX  PRINT:PRINT nar$
545 91  RETURN
546 wW  erklaerung:
547 kt  frei=1
548 Lz  WINDOW 4,,(100,0)-(631,100),0,1:GOSUB ueber
549 UJ  PRINT :PRINT " Dieses Programm, urspr"CHR$(252)"nglich f"CHR$
        (252)"r den C64 geschrieben,"
550 hI  PRINT " zeichnet eine Sternenkarte mit Sonne, Mond, Planeten
        und Fix-"
551 b1  PRINT " sternen. Wenn Sie auf einen Stern (oder in dessen N"C
        HR$(228)"he) klicken,"
552 HJ  PRINT " k"CHR$(246)"nnen Sie sich den Namen des Planeten, Ste
        rns oder des"
553 2s  PRINT " n"CHR$(228)"chsten Sternbildes nennen lassen."
554 KP  PRINT " Weiterhin besteht die M"CHR$(246)"glichkeit, den Name
        n eines bekannten"
555 ch  PRINT " Planeten (Stern/Sternbild), in Kleinbuchstaben einzug
        eben. In"
556 OA  PRINT " diesem Fall zeigt ein Pfeil dessen Position an. Bei e
        inem Stern-"
557 Pv  PRINT " bild blinken die dazugeh"CHR$(246)"rigen Sterne."
558 rk  GOTO schr
559 M7  mondphasen:
560 x6  frei=1
561 AH  WINDOW 4,,(100,0)-(631,40),0,1:GOSUB ueber
562 Jz  PRINT :PRINT " Ein Mondzyklus gliedert sich in 4 Phasen. So w
        ird z.B. ca. 3"
563 Kc  PRINT " Tage vor und auch nach Vollmond der Mond als Scheibe
        dargestellt."
564 xq  GOTO schr
565 qd  rechte:
566 3C  frei=1
567 GN  WINDOW 4,,(100,0)-(631,40),0,1:GOSUB ueber
568 1x  PRINT :PRINT "              Die Amigaversion von STERNENHIMMEL
         ist"
569 uRo PRINT "                          eine Umsetzung vom C 64"
570 3w  GOTO schr
571 tg  ueber:
572 v5  WINDOW OUTPUT 4
573 A3  PRINT :PRINT "              STERNENHIMMEL     Amiga-Version von
         G"CHR$(246)"lzy"
574 6u  LINE (2,20)-(529,20)
575 dF  RETURN
576 f1  erro:
577 QR  WINDOW 2
578 gM  RESUME pri
579 TX  fixst:
580 SK  DATA 037.8,89.3,polaris,0
581 1N  DATA 269.8,86.0,umi2,0
582 1R  DATA 252.5,82.2,umi3,0
583 ax  DATA 246.3,75.5,umi4,0
584 gF  DATA 238.5,78.0,umi5,0
585 XR  DATA 230.2,71.8,pherkad,0
586 hQ  DATA 222.7,74.2,kochab,0
587 M1  DATA 206.9,49.3,benetnasch,1
588 Kx  DATA 201.0,54.9,mizar,1
589 EB  DATA 193.5,56.0,alioth,1
590 It  DATA 183.9,57.0,megrez,1
591 sN  DATA 178.5,53.7,phekda,1
592 CQ  DATA 165.9,61.8,dubhe,1
593 vQ  DATA 165.5,56.4,merak,1
594 Gy  DATA 28.6,63.7,cas1,2
595 Or  DATA 21.5,60.2,cas2,2
596 1J  DATA 14.2,60.7,cas3,2
597 vR  DATA 10.1,56.5,schedir,2
598 Kn  DATA 2.3,59.2,caph,2
599 yq  DATA 3.3,15.2,algenib,3
600 XW  DATA 346.2,15.2,markab,3
601 DV  DATA 345.9,28.1,scheat,3
602 e5  DATA 311.6,34.0,cyg1,4
603 oi  DATA 310.4,45.3,deneb,4
604 On  DATA 305.6,40.3,schedir,4
605 U9  DATA 296.2,45.1,cyg4,4
606 9r  DATA 292.7,28.0,albireo,4
607 OZ  DATA 302.8,-0.8,aql1,5
608 Jq  DATA 297.7,08.9,atair,5
609 he  DATA 296.6,10.6,aql3,5
610 NW  DATA 292.0,03.5,agl4,5
611 Eq  DATA 286.6,-4.9,aql5,5
612 HO  DATA 286.4,13.9,aql6,5
613 Om  DATA 284.7,32.7,lyr1,6
614 Ph  DATA 283.0,37.0,lyr2,6
615 4d  DATA 282.5,33.4,lyr3,6
616 Tx  DATA 280.2,38.8,lyr4,6
617 J7  DATA 279.2,38.8,wega,6
618 jp  DATA 265.6,-39.0,sco1,7
619 OD  DATA 264.3,-43.0,sco2,7
620 Na  DATA 263.4,-37.1,sco3,7
621 il  DATA 262.7,-37.3,sco4,7
622 HP  DATA 252.5,-34.3,sc5,7
623 T4  DATA 249.0,-28.2,sco6,7
624 6X  DATA 247.3,-26.4,antares,7
625 rR  DATA 241.4,-19.8,acrab,7
626 mB  DATA 240.1,-22.6,sco9,7
627 9P  DATA 239.7,-26.1,sco10,7
628 CM  DATA 228.9,33.3,boo1,8
629 sg  DATA 225.0,41.0,boo2,8
630 Jr  DATA 221.2,27.1,boo3,8
631 wY  DATA 218.0,38.3,boo4,8
632 J1  DATA 218.0,30.5,boo5,8
633 Ho  DATA 213.9,19.2,arktur,8
634 eG  DATA 201.3,-11.2,spika,9
635 sm  DATA 198.0,-7.5,vir2,9
636 bM  DATA 195.5,11.0,vir,9
637 kw  DATA 193.9,3.4,vir4,9
638 pd  DATA 190.0,-1.0,vir5,9
639 g0  DATA 185.0,0.0,vir6,9
640 xJ  DATA 177.0,2.0,vir7,9
641 G2  DATA 191.9,-59.7,cru1,10
642 CO  DATA 187.8,-57.1,cru2,10
643 pe  DATA 186.7,-63.1,cru3,10
644 LF  DATA 183.8,-58.8,cru4,10
645 gR  DATA 177.3,14.5,denebola,11
646 Bc  DATA 168.6,15.4,leo2,11
647 1K  DATA 168.5,20.5,leo3,11
648 Kh  DATA 155.0,19.9,leo4,11
649 dh  DATA 152.1,12.0,regulus,11
650 sU  DATA 151.8,16.8,leo7,11
651 dX  DATA 146.5,23.8,leo6,11
652 Gt  DATA 116.3,28.0,pollux,12
653 e0  DATA 113.7,31.9,kastor,12
654 KD  DATA 101.3,12.9,gem3,12
655 cD  DATA 101.0,25.1,gem4,12
656 DJ  DATA 99.4,16.4,gem5,12
657 HL  DATA 95.7,22.5,gem6,12
658 rE  DATA 111.0,-29.3,cma1,13
659 8V  DATA 107.1,-26.4,cma2,13
660 gc  DATA 104.7,-29.0,cma3,13
661 c7  DATA 101.3,-16.7,sirius,13
662 1m  DATA 98.2,-18.0,cma5,13
663 CJ  DATA 90.0,37.2,aur1,14
664 CY  DATA 89.9,45.0,aur2,14
665 6Z  DATA 79.2,46.0,capella,14
666 OJ  DATA 75.5,43.8,aur4,14
667 1x  DATA 74.3,33.2,aur5,14
668 UV  DATA 88.8,7.4,beteigeuze,15
669 xH  DATA 86.9,-9.1,ori2,15
670 16  DATA 85.2,-2.0,ori3,15
671 6B  DATA 84.1,-1.2,ori4,15
672 6A  DATA 83.0,-0.3,ori5,15
673 BT  DATA 81.3,6.4,bellatrix,15
674 Jm  DATA 78.6,-8.2,rigel,15
675 Rb  DATA 59.5,40.0,per1,16
676 EX  DATA 58.5,31.9,per2,16
677 az  DATA 55.7,47.8,per3,16
678 s4  DATA 51.1,49.9,algenib,16
679 NB  DATA 47.0,41.0,algol,16
```

Listing. »Sternenhimmel« zeigt Ihnen die Position aller Himmelskörper auf dem Bildschirm an. (Fortsetzung)

```
680 5M  DATA 46.2,53.5,per6,16
681 qM  DATA 31.0,42.0,alamak,17
682 TI  DATA 17.4,35.6,mirach,17
683 7c  DATA 9.8,31.9,and3,17
684 y5  DATA 2.1,29.1,sirrah,17
685 Ng  DATA 354.8,77.6,cep1,18
686 K4  DATA 342.5,66.0,cep2,18
687 dJ  DATA 322.2,70.6,alfrik,18
688 bF  DATA 319.6,62.6,alderamin,18
689 M4  DATA 269.2,51.5,dra1,19
690 Hh  DATA 262.6,52.3,dra2,19
691 DK  DATA 231.2,59.0,dra,19
692 Gq  DATA 246.0,61.5,dra4,19
693 8U  DATA 257.2,65.7,dra5,19
694 Uw  DATA 288.1,67.7,dra6,19
695 Od  DATA 31.8,23.5,hamal,20
696 lk  DATA 28.7,20.8,ari2,20
697 dk  DATA 84.4,21.2,tau1,21
698 nN  DATA 81.6,28.6,elnath,21
699 vz  DATA 69.0,16.5,aldebaran,21
700 bt  DATA 58.8,12.5,tau4,21
701 gS  DATA 56.9,24.1,plejaden,21
702 jF  DATA 114.8,5.2,prokyon,22
703 ql  DATA 111.8,8.3,cmi2,22
704 LI  sternbilder:
705 4z  DATA ursar minor,kleiner wagen,6,1
706 mA  DATA ursar maior, grosser wagen,6,8
707 od  DATA cassiopeia,kassiopeia,4,15
708 LI  DATA pegasus,pegasus,2,20
709 81  DATA cygnus,schwan,4,23
710 32  DATA aquilla,adler,5,28
711 I6  DATA lyra,leier,4,34
712 X1  DATA scorpius,skorpion,9 ,39
713 g1  DATA bootes,bootes,5,49
714 N4  DATA virgio,jungfrau,6,55
715 HZ  DATA crux,kreuz sueden,3,62
716 F1  DATA leo,loewe,6,66
717 Am  DATA gemini,zwillinge,5,73
718 jC  DATA canis maior,grosser hund,4,79
719 3c  DATA auriga,fuhrmann,4,84
720 BP  DATA orion,orion,6,89
721 yJ  DATA perseus,perseus,5,96
722 04  DATA andromeda,andromeda,3,102
723 xA  DATA cepheus,kepheus,3,106
724 FM  DATA draco,drache,5,110
725 Vo  DATA aries,widder,1,116
726 mV  DATA taurus,stier,4,118
727 pO  DATA canis minor,kleiner hund,1,123
728 8H  planet:
729 4n  DATA uranus,0.0117,98.38,170.173
730 sA  DATA 5.70,.0472,73.847,0.773,19.1823
731 c9  DATA saturn,0.0335,158.36,92.460
732 aE  DATA 5.50,.0557,113.511,2.486,9.5810
733 X4  DATA jupiter,.0831,316.19,13.839
734 9e  DATA 5.30,.0485,100.146,1.305,5.2028
735 yY  DATA mars,0.5240,144.14,335.507
736 9a  DATA 11.00,.0934,49.326,1.850,1.2237
737 pS  DATA venus,1.6021,80.85,131.149
738 Lx  DATA 0.76,.0068,76.410,3.394,0.7233
739 BZ  DATA merkur,4.0923,31.9,76.987
740 iC  DATA 23.00,.2056,47.826,7.004,0.3871
741 Sg  pfeil:
742 nX  DATA 0,0,0,0,0,0,0,0,0,0,2,0,0,0,16,0,0,0,13,0,25,0,3,0,0,1
        92,0,224,0,120
743 Jq  DATA 0,126,0,63,128,62,0,7,0,3,128,1,192,0,0,0,0,0,0,0,192,
        0,224,0,248,0
744 nB  DATA 254,0,127,128,126,0,127,0,15,128,3,192,3,192,0,0,0,0,0,0
        ,0,255,0,0,15,128
745 6Y  mondat:
746 jn  DATA 0,31,59,90,120,151,181,212,243,273,304,334
(C) 1987 M&T
```

Listing. »Sternenhimmel« zeigt Ihnen die Position aller Himmelskörper auf dem Bildschirm an. (Schluß)



Guru Meditation

Kudiplo, Ausgabe 10-'87, Seite 75:

Hier haben sich mehrere kleine Fehler eingeschlichen.

In Zeile 145 fehlt vor

```
/xstep/2"
```

eine Klammer (»)«). Das Wort »diskussion« in Zeile 128 muß natürlich mit zwei »s« geschrieben werden. In den Zeilen 207 und 208 muß es immer

```
ein$
```

und nicht »in$« oder »eins$« heißen. Das letzte Komma in der Zeile 232 muß entfernt werden. In Zeile 515 müssen Sie die Zeilennummer 1987 einfügen, damit das Programm läuft.

Clouds, Ausgabe 10-'87, Seite 91:

Die Zeile 4 muß lauten:

```
SCREEN 2,320,267,5,1
```

Epson FX/RX-80 am Amiga, Ausgabe 11-'78, Tips und Tricks, Seite 81:

Die Zeile 10 muß lauten:

```
10 treiberfile$ = "df0:
   devs/printers/Epson"
```

Außerdem fehlt zwischen den Zeilen 110 und 140 eine OPEN-Anweisung:

```
135 OPEN "R",1,
    treiberfile$+
    "_1920",1
```

Eine Korrektur, die Public Domain-Seite betreffend:

Drei der insgesamt 102 Fish-Disks enthalten teilweise Programme, die keine Freesoft oder Shareware sind. Diese drei Disketten wurden von Fred Fish bearbeitet, die »verbotenen« Programme entfernt. Im Klartext bedeutet das, daß die Fish-Disks 57, 80 und 88 nicht mehr angeboten werden und von den Besitzern gelöscht werden sollen. Ersatz für diese Disketten sind die Update-Fish-Disks 97 (für 57), 90 (für 80) und 89 (für 88), bei denen die »gefährlichen« Programme entfernt wurden (Quelle: Amiga User Group).

Hier gibt's Clubs

Chipsi Computer Club
Jasomirgottstr. 3, Mezzanin 6, A-1100 Wien.

Neben mehreren Amiga-Freaks sind auch Besitzer anderer Computer in diesem Club. Einsteigern wird verstärkt geholfen.

Comodorus
Marderstraat 72, 2000 Hamburg 65.

Club von Umsteigern vom C 64 auf Amiga. Verbindungen zu anderen internationalen Clubs sogar im Ostblock sind vorhanden. Ein Public Domain-Softwarepool ist im Aufbau. Veröffentlichung einer eigenen Clubzeitschrift. Beitrag: 10 Mark (darin sind sechs Clubzeitschriften kostenlos enthalten). Probeexemplar gegen 1,10 Mark in Briefmarken.

Computer Club Nordkirchen e.V.
Holtweg 22. 4717 Nordkirchen, Tel. 02596/2258.

Computerbesitzer aus dem Bereich Commodore, IBM-PC/XT, Spectrum und Atari finden Aufnahme. Die Leistungen sind: Hilfe für Anfänger, monatliche Treffen, Programmbibliothek, Bücherliste, quartalsweises Clubinfo. Jeden zweiten Samstag im Monat findet ein Clubtreffen zum Erfahrungsaustausch statt.

Public Project
Dechant-Röper-Str. 32, 5750 Menden 1, Tel. 02373/10225 oder 5336.

Es wird alles geboten für Amiga und Schneider CPC: monatliche Clubzeitschrift, Sammelbestellungen, Erfahrungsaustausch und gemeinsamer Vertrieb selbsterstellter Software. Clubgebühr: 10 Mark vierteljährlich.

Commodore Mini Club
Bruckersche Straße 11, 4150 Krefeld 29, Tel. 02151/735946.

Der Club mit etwa 140 Mitgliedern befaßt sich mit allen Commodore-Computern (VC 20 bis Amiga und PCs). Geboten wird für eine Jahresgebühr von 35 Mark eine Clubzeitschrift, Public Domain-Bibliothek, Hardware-Service, Händlerkontakte und Computerbörsen. Für Kontakte steht auch die Hexagon Mailbox zur Verfügung: 8N1, 02162/58457.

PUBLIC DOMAIN SOFTWARE

180 Disk. Amiga je DM 8,-
günstige Staffelpreise und Satzpreis
ca. 1200 Disk. MS-DOS 5,25″ je DM 8,-
3,5″ je DM 10,-
Liste anfordern gegen Rückporto (0,80 DM in Marken), Computertyp angeben.
Zweitlaufwerk anschlußfertig für Amiga DM 360,-

Leerdisketten, Spitzenqualität von SENTINEL, neutral verpackt.
10 Stück DM 29,- 100 Stück DM 270,-
Viele weitere Angebote auf Anfrage.

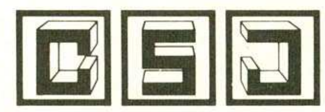
Kopierservice Public Domain Software
Dipl.-Betriebswirt Christian Bellingrath
Hans-Böckler-Str. 55 · 5860 Iserlohn · Tel. 02371/24192 · Tlx. 827937

AMIGA:

Die Massenspeicher kommen...

20-MB-Festplatte, anschlußfertig .... DM 1398,-
40-MB-Festplatte, anschlußfertig .. ab DM 2498,-
70-MB-Festplatte, anschlußfertig ..... DM 4198,-
Weitere Größen auf Anfrage!

P M D 08106/33941
Matthias Aures · Postfach 100105 · 8011 Vaterstetten



COMPUTERSOFT JONIGK

AMIGA-SPIELE

ALIEN FIRES	79,90	SHANGHAI	79,90
ARCTICFOX	84,90	SINBAD	86,—
BALANCE OF POWER	89,90	SORCERER	84,90
BORROWED TIME	79,90	SPACE BATTLE	29,95
BRIAN CLOUGH FOOTBALL	79,90	STARCROSS	84,90
CITY DEFENCE	29,95	STARGLIDER	84,90
CHAMPIONSHIP FOOTBALL	79,90	STATIONFALL	84,90
CHESSMASTER 2000	129,—	STRIP POKER	29,95
CHALLENGER	29,95	SWOOPER	49,—
CRUNCHER FACTORY	29,95	TURBO CAR	64,90
CRIMSON CROWN	84,90	TERRORPODS	79,90
DEEP SPACE	96,90	THE GUIDS OF THIEVES	79,90
DEFENDER OF THE CROWN	86,—	THE HALLY PROJECT	79,90
DEJA VU	89,90	THE PAWN	79,90
DEMOLOTION	29,95	ULTIMA III	79,90
DR.FRUIT	29,95	UNINVITED	84,90
EMERALD MINE	29,95	WINNIE THE POO	74,90
SUPER HUEY	69,—	WILLY THE KID	29,95

PREISHITS DES MONATS

PHANTASIE III	**59,-**	**CALIFORNIA GAMES**	**69,-**
KAMPFGRUPPE	**89,-**	**IMPACT**	**49,-**
BAD CAT	**64,-**	**DIABLO**	**64,-**

		AMIGA-ANWENDER	
FORTRESS UNDERGROUND	29,95		
GALAXION FIGHT	98,90	AEGIS ANIMATOR & IMAGES	298,—
GOKART RACING	29,95	AEGIS SONIX	189,—
GARRISSION	59,—	AEGIS IMPACT	439,—
GOLD RUNNER	79,90	AEGIS DIGA	149,—
KNIGHT ORC	59,90	AEGIS VIDEOSCAPE 3D	359,—
KARATE KID II	79,90	DRUM STUDIO (DEUTSCH)	79,—
KARATE KING	49,95	GISMOZ	198,—
LEATHER GODESS	89,90	LATTICE C COMPILER V3.10	498,—
MOUSE TRAP	49,90	MARAUDER II	98,—
MINDSHADOW	79,90	METACOMCO CAMBRIDGE LISP	498,—
PLUTOS	49,90	NEW ART MONITOR (DEUTSCH)	79,—
RODWAR 2000	79,90	SUPERBASE (DEUTSCH)	249,—
RACTER (SPRACHVERSION)	84,90	VIDEODIGITIZER DIGI VIEW	498,—
ROCKET ATTACK	29,95	VIZAWRITE AMIGA	198,—

★ WIR SUCHEN NOCH PROGRAMMAUTOREN ★ ★ PREISÄNDERUNGEN VORBEHALTEN ★



CSJ COMPUTERSOFT JONIGK ★ HÄNDLERANFRAGEN ERWÜNSCHT ★
An der Tiefenriede 27 ★ 3000 Hannover 1 ★ Tel. Bestellservice (0511) 886383
Riesenauswahl an Software ★ sofort CSJ NEWS anfordern (Computertyp ang.) ★
Versand Inland: Vorkasse + 2,50 DM (Euroscheck in DM): per Nachnahme + 7,- DM

MFS

Hard - & Software
MFS Hard- & Software, tel. 037/ 36 20 60, Postfach 78, 3185 Schmitten

Proudly presents

Bereits erhältlich sind folgende Prg´s für A500/1000/2000:

EFFECT CREATOR
Stellt Ihnen über 100 Effekte aus allen Breichen zur Verfügung. Mit dem EFFEKT EDITOR können Sie die Sounds editieren und Dank den Sources in PRG einbauen.
DM 99,—
sFR 79,—

LEARN ENGLISH
ist ein Englisch- und Vokabel Trainer. Wortschatz selbstständig erweiterbar.
DM 99,—
sFR 79,—

QUICK FAKTO
Einfaches FAKTURIEREN, komplettes KALKULIEREN. NORMANPASSUNG für CH/BRD/Ö. Alle Drucker ansteuerbar.
DM 99,—
nur sFR 79,—

TURBO-DRUMMER
4-SPUR-REALTIME-SEQUENCER mit Basslauf, 180 SOUNDS und vielen EFFEKTEN. Auch für ATARI ST. 2 Disketten & dt. Anl. & Kassette.
DM 149,—
nur sFR 129,—

TOOLBOX I
Viee nützliche Utilities wie: BOOTWRITER für Bootintros mit Laufschrift. SPRITE-CONVERTER und SPRITE-EDITOR. DPAINT ARTS SUPERGRAFIK.
DM 99,—
sFR 79,—

LET´S JODEL
4-SPUR-REALTIME-JODEL-SEQUENCER. Kreieren Sie Ihren eigenen Jodel und Stimmung.
DM 119,—
nur sFR 99,—

!!CRAZY CARS !!

Ein verrücktes AUTORENNEN. Jetzt DIREKT aus Frankreich mit deutscher Anleitung. >>>>>>>>>>>>>>>>>>>>>>>>>>>> **DM 89,—, sFR 69,—**

Für den AMIGA

100 % kompatibles
NEC 1036 a $3\frac{1}{2}$″-Laufwerk.
Amiga-farbenes Gehäuse.
Durchgeführter BUS.
Ein Ausschalter. **sFR. 369,—**

Auch als **Einschubdrive** für A 2000 erhältlich **sFR. 299,—**

$5\frac{1}{4}$″-Laufwerk
mit 40/80-Track-Umschaltung
Mit Busdurchführung, 880 KB.
sFR. 479,—

500er-Peripherieadapter
für 1000er-Peripherie am 500er.
sFR. 59,90

Der AMIGA-EPROMMER
brennt 2716 - 27011 mit Standard und Fast-Algorithmus.
sFR. 298,—

Software und Kabel dazu
sFR. 98,—

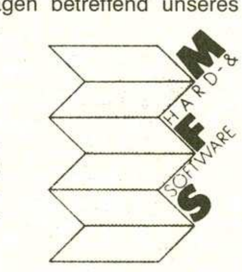
EPROMKARTE 512 KB
In 64 KB-Schritten aufrüstbar.
Durchgeschleifter Bus.
sFR. 98,—

Für AKTUELLES und andere NEUHEITEN, wie z.B. das FINAL CARTRIDGE III ist die M.F.S.-

HOTLINE

für Sie täglich unter der Telefonnumer 037/ 36 20 60 offen !!!

Im Bereich AMIGA-Soft- und Hardware sind wir bestrebt eine große Palette von qualitativ auserlesenen Produkten anbieten zu können. M.F.S. Hard- & Software ist Alleinvertreiber von Softwareprodukten der Firma Swiss Computer Arts. Dieses junge, sehr kreativ arbeitende Team ermöglichte es uns, in nächster Zeit wieder neue Software für den AMIGA in unser Angebot aufnehmen zu können.
Für spezielle Informationen oder Unterlagen betreffend unseres AMIGA 500/1000/2000-Angebotes kontaktieren sie bitte:



M.F.S. Hard- & Software,
Postfach 78, CH-3185 Schmitten

Hardware wird nur in der Schweiz verkauft. Lieferbedingungen: Lieferung am gleichen Tag, gegen Vorauskasse (+ sFR. 4,—, DM 5,—) oder Nachnahme (sFR. 6,—, DM 7,—). Gratis Gesamtkatalog anfordern, Händleranfragen erwünscht.

Tips und Tricks

»Ein Jahrmarkt der Ideen«. Hier finden Sie Antworten zu vielen Fragen. Einige Themen dieser Ausgabe: CLI, Amiga-Basic, Schutz vor Viren, Zweitlaufwerke und Icons. Sind Sie interessiert, die Geheimnisse des Amiga kennenzulernen? Schauen Sie sich um, kommen Sie herein, es sollte auch für Sie ein Tip dabei sein.

Hilfe für alle Amigabesitzer, das ist das Motto der »Tips und Tricks«. Diese Unterstützung von Lesern für Leser ist sowohl für Einsteiger als auch für Fortgeschrittene gedacht. Selbst die ausgebufftesten Profis finden hier die eine oder andere Methode, ein Problem leichter zu lösen. Viele Ratschläge helfen den Neulingen in Sachen Amiga, die ersten Hürden im Umgang mit ihrem Computer zu meistern.

Wer schon ein wenig Erfahrung mit der Programmierung und Bedienung des Amiga gesammelt hat, der wird einige Tricks schon kennen. Eventuell haben Sie auch den einen oder anderen mühsam allein ausgeknobelt. Das soll nicht bedeuten, daß Sie, sobald Schwierigkeiten auftauchen, warten sollen, bis Ihnen ein anderer Leser einen Wink gibt. Im Gegenteil, wenn Sie etwas ausgetüftelt haben, schicken Sie Ihren Tip ein. Zögern Sie nicht, bis Ihnen jemand zuvorkommt. Sie können vielen Lesern helfen. Es kann sich um jeden Trick im Umgang mit dem Amiga handeln: Hardware, Software, Programmierung, Zubehör. Schicken Sie auch Lösungen zu alten Problemen, wenn Sie glauben eine bessere Lösung gefunden zu haben. Das ist besonders reizvoll. Auch in dieser Ausgabe sind einige Tips, die auf Fragen aus dem Leserforum oder »Tips und Tricks« der vorherigen Ausgaben des Amiga-Magazins eingehen.

Mehr Speicher für Amiga-Basic

Mit dem Basic-Kommando CLEAR können Sie den verfügbaren Programm- und Variablenspeicher neu dimensionieren (siehe auch Amiga-Basic-Handbuch). Dennoch gibt es Schwierigkeiten mit dem Befehl:

CLEAR liefert häufig keinen Block, obwohl FRE(-1) noch genügend Platz anzeigt. Die Erklärung können Sie leicht experimentell ermitteln:

Laden Sie vor Aufruf des Basic-Interpreters das Utility »Free Memory« von der »ExtrasD-Diskette«. Dieses Programm öffnet am unteren Bildrand ein Fenster und gibt eine grafische Darstellung der momentanen Speicherbelegung. Laden Sie nun Amiga-Basic und Ihr Testprogramm. Das Zusatzfenster zeigt wie der freie Systemspeicher zusammenschrumpft.

Starten Sie ein Basic-Programm, das am Anfang eine CLEAR-Anweisung ausführt. Diese soll beispielsweise das Daten- und Programmsegment der voreingestellten 25000 Bytes vergrößern.

```
CLEAR , 50000
```

Das geht nur dann gut, wenn im Speicher ein weiterer zusammenhängender Block der angeforderten Größe neben dem bereits bestehenden Bereich zur Verfügung steht. »Free Memory« zeigt Ihnen sehr eindrucksvoll, wie das aktuelle Datensegment bei diesen Operationen hin- und hergeschaltet wird. Erst wird ein neuer Block reserviert, dann das alte Segment freigegeben. Das Ergebnis dieser Versuche lautet:
— Der momentane Programm- und Datenspeicher wird bei der Ermittlung eines neuen nicht einbezogen. Dies gilt auch, wenn der aktuelle Speicher nur zum Teil genutzt wird.
— Der neue Speicherbereich muß zusammenhängend sein.

Es gibt zwei Wege, um ein möglichst großes Datensegment zugewiesen zu bekommen:
1. Im Direkt-Modus verkleinern Sie das Datensegment soweit wie möglich:

```
CLEAR , 1024
```

Nun können Sie ein großes wählen. Zum Beispiel:

```
CLEAR , 100000
```

2. Im Programm erfolgt die Reservierung folgendermaßen.

```
CLEAR , 25000 - FRE(0)
REM tatsächliche Programmlänge bleibt reserviert
CLEAR, FRE(-1) -30000
REM verbleibender Speicher abzüglich einer Reserve
```

Die Reserve, die Sie dem Betriebssystem zur Verfügung stellen müssen, hängt davon ab, ob Sie im Programm Befehle verwenden, die Systemspeicher anfordern:
— SOUND
— WAVE
— WINDOW
— SCREEN
— LIBRARY

Besonders das Öffnen eines neuen Screens läßt den verfügbaren Speicher rasch dahinschmelzen.

(D. Siedentop/ub)

Fensternamen

In der »Intuition.library« gibt es eine Funktion namens »SetWindowTitles«. Sie ändert den Namen eines Fensters. Mit einer Unterroutine läßt sich dies von Basic aus realisieren:

```
SUB Titel (WT$,ST$) STATIC
  WT$=WT$+CHR$(0) : ST$=ST$+CHR$(0)
  CALL SetWindowTitles
(WINDOW(7),SADD(WT$),SADD(ST$))
END SUB
```

Die Subroutine ändert sowohl den Namen des aktuellen Fensters als auch des zugehörigen Screens. Übergeben Sie die neuen Namen beim Aufruf:

```
CALL Titel ("Neuer Fenstername", "Neuer
Screentitel")
```

Beachten Sie, daß Sie zuvor die »Intuition.library« mit dem Befehl »LIBRARY "intuition.library"« öffnen müssen.

(Tobias Helge Kosuch/ub)

Startdisketten verändern

Es ist sicher vielen Amiga-Freunden bekannt, daß man eine Startdiskette so manipulieren kann, daß beim Booten dieser Diskette eine Meldung, etwa eine Information über das Programm, eine Begrüßung oder ein Datum auf dem Bildschirm ausgegeben wird. Hierzu sind bekanntlich mehrere Schritte durchzuführen:
1. Kopieren der benötigten CLI-Befehle von einer Workbench ins c-Directory der betreffenden Startdiskette.
2. Mit dem Editor ED einen Text erstellen, der beim Booten ausgegeben werden soll und diesen auf die Startdiskette speichern.
3. Ändern der Startup-Sequence, damit beim Booten die Startmeldung ausgegeben wird.

Wer nur ein Laufwerk besitzt und häufig auf diese Art seine Disketten manipuliert, wird bald an der Tipp- und »Diskjockey«-Arbeit verzweifeln.

Mit dem Programm »announcer« wird diese Arbeit zum Kinderspiel. Das Programm nimmt Ihnen zwar nicht Arbeit zum Erstellen des gewünschten Textes ab, aber die Änderungen der Startup-Sequence und des c-Directories werden wesentlich vereinfacht. Sie schreiben die gewünschte Startmeldung in eine Textdatei in die RAM-Disk, legen die Diskette ein und tippen:

```
EXECUTE ann name
```

Die Diskette wird dann automatisch mit allen nötigen Änderungen versehen. (Anmerkung der Redaktion: Im Zweifelsfall immer mit einer Sicherheitskopie arbeiten!)

Sie müssen allerdings eine neue Workbench- beziehungsweise CLI-Diskette erstellen, mit deren Hilfe sich beliebige Startdisketten einfach manipulieren lassen. Starten Sie zunächst mit einer Workbench-Kopie Ihren Amiga und rufen das CLI auf. Im CLI führen Sie folgende Schritte aus:

1. Rufen Sie den Editor mit

```
ED startup-s.zusatz
```

auf, um folgendes kurzes Text-File zu schreiben:

```
TYPE s/starttext
WAIT 15
```

(Verlassen des Editors mit <ESC x>)

2. Geben Sie ebenfalls mit dem Editor das Batch-File mit dem Namen »ann« ein. Der Aufruf lautet:

```
ED ann
```

```
.key name/a
IF EXISTS RAM:<name>
ECHO "Start von announcer"
CD df0:
COPY RAM:c/type to df0:c/type
COPY RAM:c/wait to df0:c/wait
COPY RAM:<name> to df0:s/starttext
RENAME s/startup-sequence TO s/oldss-seq
JOIN RAM:startup-s.zusatz s/oldss-seq AS df0:s/
startup-sequence
ECHO "ende von announcer"
ECHO "neue Diskette einlegen!"
ELSE
ECHO "Datei ram:<name> nicht gefunden"
ENDIF
```

3. Nun müssen Sie mit dem Editor die Startup-Sequence der Workbench ändern. Fügen Sie hierzu die folgenden, nicht mit »*« markierten Zeilen ein:

```
* IF EXISTS sys:utilities
* PATH sys:utilities ADD
* ENDIF

CD RAM:
MAKEDIR RAM:c
COPY df0:c TO RAM:c
ASSIGN c: RAM:c
COPY df0:startup-s.zusatz TO RAM:
COPY df0:ann to RAM:
ASSIGN announcer to RAM:ann

* SETMAP d
```

Erklärung des Programms:

Alle benötigten Batch-Files, Textdateien und CLI-Befehle werden beim Starten der Workbench ins RAM kopiert. Den Text, den Sie auf eine Diskette kopieren wollen, können Sie mit dem Editor schreiben und unter einem beliebigen Namen in der RAM-Disk ablegen. Das Programm »announcer«, das Sie mit

```
EXECUTE ann name
```

aufrufen, arbeitet wie folgt:

— Die zum Ausgeben des Textes benötigten Befehle werden aus der RAM-Disk ins c-Directory der eingelegten Diskette kopiert
— Die Startup-Sequence dieser Diskette wird mit einem Befehl zur Ausgabe der Textdatei versehen
— Die alte Startup-Sequence wird vorsichtshalber unter dem Namen »oldss-seq« gerettet
— Der gewünschte Text, der sich in der RAM-Disk in der mit »name« bezeichneten Datei befinden muß, wird im s-Verzeichnis gespeichert. (Frank Zielen/ub)

Externes Laufwerk zum Booten

Mit einer Änderung der »Startup-Sequence« können Sie den Bootvorgang auf ein externes Laufwerk umleiten. Dies ist interessant für die Besitzer eines 5¼-Zoll-Laufwerks, die ein Programm auf 5¼-Zoll-Diskette kopiert haben, das sich aber nur durch Booten starten läßt.

Für die Änderung müssen Sie eine Startdiskette präparieren, die den Bootvorgang auf das externe Laufwerk umlenkt. Am einfachsten verwenden Sie eine normale Workbenchkopie.

Auf dieser ändern Sie die »Startup-Sequence«:

```
ASSIGN S:      df1:s
FAILAT 21
ASSIGN L:      df1:l
ASSIGN FONTS: df1:fonts
ASSIGN LIBS:  df1:libs
ASSIGN devs:  df1:devs
ASSIGN SYS:   df1:
ASSIGN C:     df1:c
df0:c/CD SYS:
df0:c/EXECUTE Startup-Sequence
```

Probieren Sie das Programm aus. Legen Sie Ihre »externe Bootdiskette« in Laufwerk df1: und die modifizierte Workbench im Laufwerk df0: ein. Jetzt starten Sie den Amiga erneut.

Diese Methode hilft oft aber nicht immer. Sie versagt zum Beispiel bei Programmen, die das interne Laufwerk direkt ansprechen, um dort Daten zu lesen. (Angela Schmidt/ub)

Schutz vor Viren

Wer sich durch eine Unachtsamkeit oder aus Unwissenheit einen Virus auf seine Diskette kopiert hat, kann diesem mit dem CLI-Befehl INSTALL zu Leibe rücken. Dieser Befehl überschreibt die beiden ersten Blocks auf der Diskette (Bootblock), in denen sich die meisten Viren versteckt halten. Im Bootblock befindet sich die Information für das Betriebssystem, ob eine Diskette eine Startdiskette ist, sowie ein Assemblerprogramm, das der Amiga beim Booten ausführt. INSTALL macht aus jeder formatierten Diskette eine normale Startdiskette. Der Befehl löscht also auch alle mit dem »BootBoy« erstellten Vorspänne. Jedoch seien Sie vorsichtig. Wenden Sie den Befehl INSTALL nicht wahllos auf allen Ihren Disketten an, um sich vorsorglich vor Computerviren zu schützen. Es gibt zahlreiche Programme, wie zum Beispiel »Archon«, die Ihre eigene Startroutine besitzen. Diese würden Sie unwiderruflich löschen. Das ganze Spiel wäre unbrauchbar.

(Ralf Dittmer/ub)

NewCLI mit Komfort

Dem Befehl NEWCLI kann außer den gewünschten Dimensionen des Ausgabe-Fensters auch noch ein weiterer Parameter mitgegeben werden. Dieser bezeichnet ein Batch-File, das als »Startup-Sequence« für das neue CLI-Fenster ausgeführt wird. Ein Beispiel:

```
NEWCLI CON:0/128/640/128/Mein-CLI Startsequenz
```

führt nach Eröffnung des spezifizierten Fensters sofort die Befehle aus, die im Batch-File »Startsequenz« stehen.

(Christian Cachin/ub)

Selbstgestrickte Icons schützen

Mit Hilfe des CLI-Befehls PROTECT können Sie Ihre selbsterstellten Icons schützen. Dies ist für Basic-Programmierer interessant.

Wenn Sie für Ihr Programm mühsam ein eigenes Icon erstellt haben, dann das Programm ändern und erneut speichern, löscht Amiga-Basic brutal Ihr Icon. Deshalb sollten Sie das Icon beziehungsweise die zugehörige Info-Datei mit diesem Befehl sichern:

```
PROTECT Programmtitel.info R
```

Jetzt hat Amiga-Basic keine Chance mehr. Dieses Icon kann nur noch gelesen werden. (Rolfdieter Schiedrum/ub)

Direkt in die Bitmap

Für besonders schnelle Assembler-Programme ist es oft wünschenswert, Punkte auf dem Bildschirm direkt durch Adressierung der Bitplanes anzusprechen. Eine Möglichkeit hierzu bieten die zwei Unterroutinen »PSet« und »PClear«:

```
PClear:
        move.b #$b5, label+1
PSet:
        divu     #8, d0
        mulu    #40, d1
        add.w        d0, d1
        swap         d0
        move.b   #7,  d2
        sub.b   d0,  d2
label:
        bset    d2, (a5,d1)
        move.b #$f5, label+1
        rts
```

Die Unterprogramme sind mit dem Seka-Assembler geschrieben. Um einen Punkt zu setzen oder zu löschen, rufen Sie »PSet« beziehungsweise »PClear« auf. Übergeben Sie folgende Werte in den genannten Registern:
X-Koordinate des Punktes in d0
Y-Koordinate des Punktes in d1
Zeiger auf Bitplane in a5.

Die Startadressen der Bitplanes eines Screens sind leicht zu finden. Die Intuition-Funktion »OpenScreen« liefert den Zeiger auf die Screendaten. Addieren Sie zu diesem Wert #$C0. An der berechneten Stelle befindet sich die Adresse der ersten Plane. Der Pointer auf die zweite und alle weiteren Planes liegt jeweils vier Byte höher.

Diese direkte Programmierung der Bitplanes ist etwas für Tüftler und birgt viele Fehlerquellen. Sie können Punkte auch durch einfache Angabe der Koordinaten mit Hilfe der Systemroutinen in der »graphics.library« setzen und löschen. In Basic steht Ihnen hierfür der Befehl PSET zur Verfügung. Allerdings sind diese Befehle ein wenig langsamer. (David Czock/ub)

Gleiche Icons für RAM-Disk und Workbench

Wollen Sie für die Workbench und die RAM-Disk eigene Icons verwenden? Mit einem kleinen Kniff ist dies zu schaffen.

Sobald Sie die RAM-Disk mit dem Befehl »DIR ram:« initialisieren, erscheint immer das Standard-Icon auf der Workbench, auch wenn Sie für die Workbench-Diskette bereits ein eigenes Icon entworfen haben. Möchten Sie die RAM-Disk ebenfalls durch Ihr eigenes Icon darstellen, so erreichen Sie dies mit Hilfe der Initialisierung:

```
COPY disk.info to ram:
```

Workbench-Icon und RAM-Icon sind nun gleich. Soll die RAM-Disk ein ganz besonderes Icon besitzen, so müssen Sie dieses auf der Diskette speichern und die RAM-Disk durch den Befehl:

```
COPY spezialicon.info to ram:
```

aufrufen. (Klaus Rupp/ub)

Unprotect

Eine geschützte Datei oder Icon (siehe nebenstehender Tip) kann mit »DELETE« nicht mehr gelöscht werden. Dieser Schutz ist sinnvoll, um ein unbeabsichtigtes Löschen von Dateien zu vermeiden. Wenn Sie sich aber doch einmal fest entschlossen haben, eine gesicherte Datei zu verändern, können Sie den Schreibschutz aufheben:

```
PROTECT filename RWED
```

Dieser Befehl erlaubt Ihnen den freien Zugriff auf Ihre Datei.

(Laurent Münster/ub)

Startup mit Abfrage

Kopieren Sie die CLI-Befehle nur bei Bedarf in die RAM-Disk. Die CLI-Befehle in der RAM-Disk sparen Zeit und schonen Ihre Disketten — falls Sie mit dem CLI arbeiten. Das Kopieren benötigt jedoch Speicherplatz und verlängert die Boot-Zeit. Benötigen Sie das CLI nicht, sondern möchten mit der Workbench arbeiten, so ist die Installation der RAM-Disk in der »Startup-Sequence« unnötig. Für diesen Fall können Sie sich zwei Startdisketten erstellen:

Eine ausschließlich für die Arbeit mit dem CLI, die zweite für die Workbench. Das kostet eine Diskette. Es gibt eine trickreichere Methode. Bauen Sie eine Abfrage in der »Startup-Sequence« ein. Der Trick, der hierzu angewendet wird, ist die Verwendung der Befehle FAILAT und CD:

Sie müssen Ihre Startdiskette (im allgemeinen eine Kopie der Orginal-Workbench) vorher präparieren.

1. Erzeugen Sie ein neues Dateiverzeichnis mit dem Namen »n«:

```
MAKEDIR n
```

2. In diesem Verzeichnis erzeugen Sie mit dem Editor ein File mit dem Namen »dummy«:

```
ED n/dummy
```

Es spielt keine Rolle, was in diesem File steht. Schreiben Sie einen Text und verlassen den Editor mit <ESC> <x> <RETURN>.

3. Ergänzen Sie diese Zeilen in der »Startup-Sequence«:

```
ECHO "wollen Sie im CLI bleiben j/n ?"
FAILAT 25
CD >nil: ?
IF NOT EXISTS dummy
  SKIP cli
ENDIF
LOADWB
ENDCLI >nil:
LAB cli
CD :
ECHO "wollen Sie RAMCLI aktivieren ?"
FAILAT 25
CD >nil: ?
IF EXISTS dummy
  SKIP nein
ENDIF
; Ein Beispiel um die RAM-Disk mit
; CLI-Befehlen zu aktivieren
MAKEDIR ram:c
COPY c: ram:c
PATH ram:c ADD
LAB nein
CD :
ECHO "Initialisierung beendet"
```

Wenn Sie nun Ihre Work-Disk neu starten, fragt der Amiga, ob Sie mit dem CLI und mit den CLI-Befehlen in der RAM-Disk arbeiten möchten. Die Erklärung zum Programm:

Das Dummygerät »NIL:« wird gewinnbringend genutzt. Durch Umlenken der Helpfunktion (?) auf dieses Gerät verfügt der Benutzer im CLI über eine elegante Eingabefunktion. Nach Ausgabe des Helptextes auf das nicht existierende Ausgabegerät können Sie einen Text eingeben. Dieser stellt nun das aktuelle Directory dar (CD-Funktion).

Jetzt sucht der Computer in diesem Directory nach dem File »dummy«. Findet er es, war die Eingabe »n«, da nur im Directory »n« ein File »dummy« existiert. Mit Hilfe dieses Tricks läßt sich auch eine Paßwortkontrolle realisieren. Sie brauchen nur ein Directory zu generieren, dessen Name Ihrem Codewort entspricht. In diesem Verzeichnis muß wiederum das File »dummy« stehen. Wird ein falsches Schlüsselwort eingegeben, so findet der Amiga die Datei nicht und beginnt erneut mit der Abfrage.

Dies hilft auf jeden Fall, um die neugierige Schwiegermutter vom Amiga fernzuhalten. Falls sie nicht weiß, daß man die »Startup-Sequence« mit <CTRL D> abbrechen kann.

(Martin Horn/ub)

Bildschirmlöschen in C

Das folgende kleine C-Programm löscht den Bildschirm beziehungsweise das aktuelle Fenster:

```
{
 printf("%c", 12 );
}
```

Nennen Sie das Sourcefile »CLS.c«, dann heißt Ihr compiliertes Programm »CLS«. Dieses können Sie in den C-Ordner der Workbench-Diskette kopieren. So steht Ihnen unter Amiga-DOS der Befehl CLS zur Verfügung. (Steffen Unterreiner/ub)

File-COPY mit einem Laufwerk

Mit diesem Utility können Sie mit nur einem Laufwerk komfortabel Ihre Files kopieren. Das Batch-File »FILECOPY« besteht nur aus CLI-Befehlen und arbeitet in der RAM-Disk:

```
.key file

ECHO "Bitte Source-Disk in Drive df0:"
ECHO "und ENTER drücken"
DATE >nil: ?
fail 10
cc df0:<file> ram:
IF NOT EXISTS <file>
  ECHO " Datei <file> existiert nicht"
  ECHO "Rufen Sie die Kopierroutine erneut auf"
  QUIT 10
ENDIF
ECHO " Bitte Ziel-Disk in Drive df0: "
ECHO " und ENTER drücken"
DATE >nil: ?
cc ram:<file> df0:
ECHO "ready"
```

Zur Initialisierung der Kopierroutine werden die benötigten CLI-Befehle mit Hilfe des Batch-Files »CLIRAM« in die RAM-Disk kopiert:

```
COPY df0:c/COPY     ram:cc
COPY df0:c/ECHO     ram:
COPY df0:c/FAILAT   ram:fail
COPY df0:c/IF       ram:
COPY df0:c/ENDIF    ram:
COPY df0:c/QUIT     ram:
COPY df0:c/DIR      ram:
COPY df0:c/DATE     ram:
COPY df0:c/ED       ram:
COPY df0:c/EXECUTE  ram:ex
COPY df0:c/MAKEDIR  ram:mdir
COPY df0:FILECOPY   ram:
CD ram:
DIR
```

Gehen Sie wie folgt vor: Geben Sie mit dem Editor beide Batchfiles ein und speichern diese auf Ihrer Work-Disk (Workbench-Kopie oder CLI-Diskette). Initialisieren Sie das Kopierprogramm mit dem Befehl:

```
EXECUTE CLIRAM
```

Zum Kopieren eines Files auf eine andere Diskette dient der Befehl:

```
ex FILECOPY Name
```

Das Programm fordert die benötigten Disketten automatisch an. Sie müssen nur die Disketten wechseln und mit <RETURN> quittieren. (Johannes Mayr/ub)

(Anmerkung der Redaktion: Das Batchfile speichert einige CLI-Befehle unter abgekürzten Namen in der RAM-Disk. Dies ist praktisch, erspart Tipparbeit, aber kann zu Verwechslungen führen. Grundsätzlich sollten Sie die CLI-Befehle nicht umbenennen.)

COPY, COPY, COPY

Der vorherige Tip verwendet viele COPY-Befehle. Sie können sich Tipparbeit ersparen und mehrere COPY-Zeilen in einem »Pattern« zusammenfassen. Beispielsweise durch die Anweisung:

```
COPY df0:c/(COPY| ECHO| FAILAT| IF| DIR| ) ram:c
```

(ub)

Kurze Namen für Disketten

Im CLI gibt es den Befehl RELABEL. Mit diesem ändern Sie den Namen einer Diskette. So können Sie einer Diskette einen sinnvollen und praktischen Namen geben. Eine Kopie der Workbenchdiskette beim Amiga 500 trägt beispielsweise den Namen »Copy of A500 WB 1.2«, falls das Duplikat mit Hilfe der Workbench erstellt wurde. Diese Bezeichnung ist lang und enthält mehrere Blanks. Dadurch wird der Umgang mit dieser Diskette im CLI erschwert. Diese Arbeit entfällt, wenn Sie die Diskette umbenennen:

```
RELABEL df0: "a500WB"
```

Sie können auch ganz persönliche Namen verwenden. Entscheidend ist, daß Sie Ihre Disketten am Namen erkennen und dieser einfach einzutippen ist. Der Name einer Diskette läßt sich auch über die Workbench mit Hilfe des Menüpunkts »Rename« ändern. (Reinhard Kersten/ub)

Immer auf dem rechten Weg

Haben Sie einmal vergessen, in welchem Unterverzeichnis Sie sich gerade befinden? Der CD-Befehl ohne weitere Eingaben hilft Ihnen weiter. Er gibt den Namen des vollständigen Pfades, in dem Sie gerade arbeiten, in dieser Form an:
<Diskettenname> <Unterverzeichnis1>
<Unterverzeichnis2>
Zum Beispiel:
A500WB:devs/printers
Auf diese Art können Sie sich nicht mehr in den Dateiverzeichnissen verlaufen. (Reinhard Kersten/ub)

Zweitlaufwerk braucht Speicher

Wenn Sie ein Zweitlaufwerk am Amiga anschließen, erniedrigt sich der verfügbare Speicher um etwa 20 KByte. Dieser Speicher wird als Puffer für das Laufwerk benötigt. Sollten Sie Programme laden wollen, die mit dem verfügbaren Speicher nicht auskommen, ziehen Sie das Laufwerk bei ausgeschaltetem Computer ab und versuchen erneut, das Programm zu starten.
(Hendrik Krasemann/ub)

Erweiterte Auswahl von Piktogrammen

Im Amiga Anwender-Handbuch Kapitel 4, Seite 46, finden Sie Hinweise auf die erweiterte Auswahl von Piktogrammen. Die Beschreibung dieser Option ist allerdings dürftig. Sie erfahren nur wenig, wozu diese Auswahl dient. Um eine erweiterte Auswahl zu erreichen, müssen Sie während der Wahl einzelner Piktogramme die Shift-Taste drücken. Folgende Blockoperationen sind erlaubt:
— »Discard« löscht alle angewählten Piktogramme.
— »Open« öffnet alle Schubladen, wenn kein Programm mit ausgewählt wurde.
— Sämtliche Piktogramme können durch »Ziehen« des zuletzt angeklickten Objekts in eine andere Schublade oder auf eine andere Diskette kopiert werden.
— Mit »Snapshot« werden die aktuellen Positionen jedes angeklickten Piktogramms gespeichert.

Diese gemeinsame Bearbeitung von Piktogrammen erleichtert das Arbeiten mit der Workbench. Beim Kopieren und Löschen mehrerer Objekte ist es eine wertvolle Hilfe.
(Stephan Riedelbeck/ub)

Der Pseudo-Guru

Wer ist nicht schon an den kleinen unauffälligen Fehlern verzweifelt? Wir wollen Ihren Blick für solche »Bugs« (Wanzen) in Ihrem Programm schärfen. Dazu stellen wir in jeder Ausgabe ein Programm vor, in dem Sie den Fehler finden sollen.

Es gibt Fehler, die entdeckt man ziemlich schnell, da sie angezeigt werden, zum Beispiel die berüchtigten Syntax-Fehler in Basic. Doch der Teufel steckt oft in winzigen Details, die den Computer nicht veranlassen, einen Fehler anzuzeigen. Ein solches »Fehlverhalten« des Programms zu finden ist oft mühselige Kleinarbeit. Aber auch hier macht es wie so oft die erlernte und erfahrene Routine leichter.

Zu diesem Zweck erscheint ab jetzt in jeder Ausgabe ein kurzes Programm, in dem ein kleiner, aber fataler Fehler steckt. Ihre »Hausaufgabe« soll es sein, diesen zu suchen und zu finden. In der darauffolgenden Ausgabe finden Sie die Lösung des Problems mit Erklärung, was falsch gemacht wurde.

Heute beginnen wir mit einem kleinen Basic-Programm, das eigentlich zwei Sterne auf den Bildschirm zeichnen soll, wie sie im Bild zu sehen sind.

Leider hat sich beim Programmieren ein winzig kleiner Fehler eingeschlichen.

```
ra=50
xm=100
ym=60
FOR i=0 TO 350 STEP 10
   drawstern xm,ym,i,ra
NEXT i
xm=200
FOR i=0 TO 350 STEP 20
   drawstern xm,ym,i,ra
NEXT i
WHILE MOUSE(0)=0
WEND
END
SUB drawstern (xmit,ymit,winkel,radius) STATIC
winkel=winkel*3.14159265#/180
xziel=xmit+SIN(winkel)*radius
yziel=ymit+COS(winkel)*radius
LINE (xmit,ymit)-(xziel,yziel),1
END SUB
```

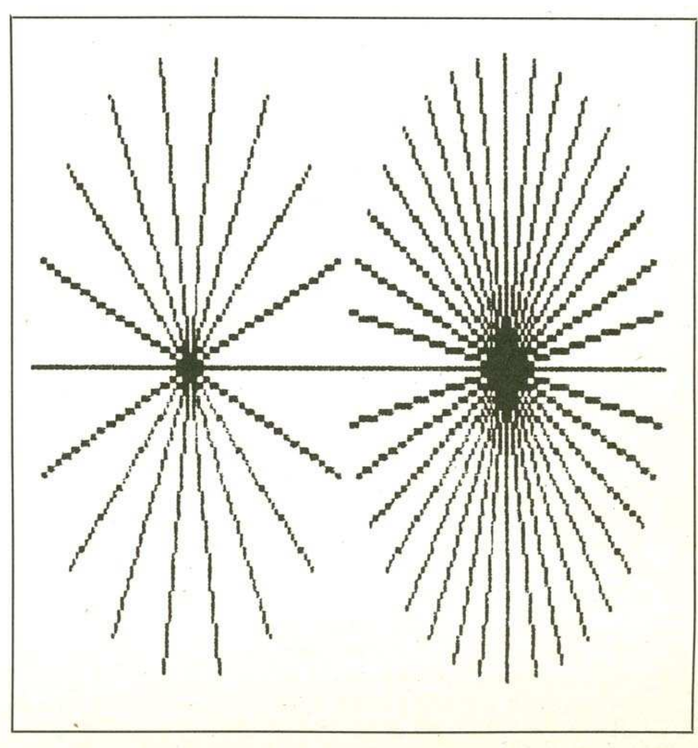


Bild. Diese Sterne sollten eigentlich gezeichnet werden

Und nun sollten Sie am besten Ihr Basic-Handbuch zu Rate ziehen, warum nicht die zwei Sterne gezeichnet werden.

Aus diesen kleinen Programmen werden Sie im Laufe der Zeit sicher lernen, viele Fehler nicht mehr zu machen. Sollten dennoch Fehler passieren, sind Sie dann sicher in der Lage, diese schneller als früher zu finden und zu beseitigen.

Und nun viel Spaß beim Knobeln und Lernen bis zum versteckten Fehler im nächsten Pseudo-Guru. (rb)

Eine Shell zum Abtippen!

Viele Amiga-Benutzer haben schon von der Metacomco-Shell gehört, mit der die Arbeit im CLI erleichtert wird. Wir bieten Ihnen so ein Programm zum Abtippen.

Das Programm »Amiga-Shell« bietet viele Funktionen, die das Arbeiten im CLI vereinfachen. Jeder Benutzer, der dort viel arbeitet, wird das Programm bald der Erleichterung wegen nicht mehr missen wollen.

Sie kennen sicher das Problem: Sie haben gerade eine lange Befehlszeile eingegeben und dabei einen Tippfehler gemacht. Mit dem normalen CLI müssen Sie jetzt den gesamten Text erneut tippen. Hier bietet die Amiga-Shell einen erheblich größeren Komfort. In der Zeile können Sie die Tasten <Cursor rechts>, <Cursor links>, <Backspace> und <Del> benutzen. Wenn Sie also am Anfang der Zeile einen Fehler gemacht haben, fahren Sie mit dem Cursor auf die entsprechende Stelle und ändern den Text entsprechend ab. Beim Editieren ist normalerweise der Einfügemodus eingeschaltet. Das heißt, daß an der Cursor-Position Text eingefügt wird. Dies läßt sich jedoch abschalten. Mit der Tab-Taste schalten Sie zwischen Überschreiben und Einfügen um.

Aber auch wenn Sie die letzte Zeile schon mit <RETURN> beendet haben, können Sie sie wieder auf den Bildschirm bekommen. Amiga-Shell merkt sich in einer Liste die letzten zwanzig Befehle, die Sie eingegeben haben. Die Befehle erreichen Sie durch Drücken der Tasten <Cursor hoch> oder <Cursor runter>. Wenn der gesuchte Befehl dann hinter dem Prompt steht, kann er natürlich noch einmal editiert werden. Um die gesamte Liste der letzten Befehle auf den Bildschirm zu bekommen, geben Sie HISTORY ein. Wenn Sie mehr als die voreingestellten zwanzig Befehle in der Liste haben wollen, übergeben Sie noch eine Zahl von 20 bis 99. Wollen Sie eine Liste mit 50 Einträgen, geben Sie einfach ein:

```
HISTORY 50
```

Ein weiterer nützlicher Befehl ist CLS, der das Fenster löscht und den Cursor in die linke obere Ecke setzt. Übrigens können Sie mehrere Befehle in einer Zeile eingeben. Dazu müssen die einzelnen Anweisungen nur durch Strichpunkte getrennt werden.

```
CLS;DIR df0:;DIR df1:
```

Diese Befehlszeile löscht zuerst den Bildschirm und gibt dann das Directory von df0: und anschließend von df1: aus.
Aber Amiga-Shell kann noch viel mehr. Durch die Belegung der Funktionstasten mit immer wieder benötigten Befehlen läßt sich viel Zeit und Arbeit sparen. Der Befehl zur Belegung ist SET. Ein denkbarer Aufruf ist:

```
SET f1 DIR df0:
```

Danach bewirkt ein Druck auf die Taste <F1>, daß in der aktuellen Zeile der Text »DIR df0:« erscheint. Natürlich können Sie diesen Text beliebig editieren. Mit diesem Befehl können Sie die Funktionstasten von <F1> bis <F10> mit Text belegen. Dabei hat die Funktionstaste <F10> die Bezeichnung »f0«. Zusätzlich zu den zehn »normalen« Funktionstasten können Sie noch die Funktionstasten mit <Shift> zusammen definieren. Dann wird beim Aufruf von SET aber das »f« groß geschrieben:

```
SET F5 info
```

Wenn Sie jetzt <Shift> <F5> drücken, erscheint der Text »info«.
Ein sehr mächtiger Befehl ist ALIAS. Er ersetzt vorgegebene Texte durch beliebige andere.

```
ALIAS gotoram ASSIGN c: ram:c;CD ram:
```

Wenn Sie nach dieser Definition »gotoram« eingeben, werden die zwei Befehle (ASSIGN und CD) durchgeführt. Dadurch lassen sich lange, immer wiederkehrende Befehlssequenzen sehr gut abkürzen. Ein besonderes ALIAS gibt es auch noch, nämlich den für »_Prompt«. Mit

```
ALIAS _Prompt Hallo:
```

erreichen Sie, daß in Zukunft das Prompt der Shell (normalerweise das Größerzeichen) verändert wird. Vor jeder Eingabezeile erscheint also nun »Hallo:«.

Doch mit ALIAS lassen sich viel erstaunlichere Vorgänge auslösen. Durch die zwei folgenden ALIAS-Anweisungen kann man viel Arbeit und Denken sparen. Es ist oft so, daß man längere Zeit mit einer Datei arbeitet und diese öfters ausdrucken will:

```
ALIAS setname DIR df0:;ECHO "Dateiname:";INPUT datei
```

Diese Sequenz zeigt den Inhalt der Diskette df0: an, gibt den Text »Dateiname:« aus und erwartet die Eingabe eines Dateinamens. So weit so gut, sinnvoll wird das Ganze aber erst durch folgenden Befehl:

```
ALIAS print RUN COPY Xfile  prt:
```

Das »X« in »Xfile« steht für ein umgekehrtes Ausrufezeichen, das man mit <Alt> <i> erreicht. Wenn Sie nun mit »setname« einen Namen eingesetzt haben, können Sie mit »print« erreichen, daß die Datei mit dem eingegebenen Namen ausgedruckt wird. Man kann also einen Aliastext in einen anderen einfügen, in dem man das umgekehrte Ausrufezeichen davorsetzt.

Eine bestimmte Art der Zuweisung dürfen Sie allerdings nicht benützen, wenn Sie keinen Wert auf eine Guru-Meditation legen.

```
ALIAS a Xa
```

Das X steht wieder für <Alt> <i>. Hierbei ruft sich »a« nämlich immer wieder selbst auf. Um eine Liste aller Aliasnamen zu erhalten, geben Sie ALIAS ohne Parameter ein.

Wer viel arbeitet, wechselt auch oft das Unterverzeichnis, in dem er sich befindet. Im normalen CLI benötigt man dazu den Befehl »CD«. Amiga-Shell erleichtert auch hier die Arbeit, da sie selbständig erkennt, ob es sich um eine Datei oder ein Directory handelt. Bei Eingabe von

```
ram:
```

wechselt man das Verzeichnis und befindet sich dann auf der RAM-Disk. Natürlich können hier auch längere Pfadnamen verwendet werden.

Viel Komfort im CLI

Für den CLI-Befehl EXECUTE gibt es eine Entsprechung für die Amiga-Shell: »exec«. Sie funktioniert genau wie der alte Befehl, nur kann sie auch die Befehle der Amiga-Shell durchführen. Somit kann man eine eigene Startup-Sequence für die Amiga-Shell erstellen. In dieser Datei können dann Tastenbelegungen und Aliasdefinitionen stehen.

Für diejenigen, die aus dem Programm noch lernen wollen, folgt nun noch eine Erläuterung zum Programm selbst. Nach dem Initialisieren der Variablen prüft das Programm, ob im Programmaufruf noch der Name einer Startup-Sequence angegeben ist. Ist das der Fall, wird die Routine Execfile aufgerufen. Jetzt kommt das Programm in die Hauptschleife. In ihr werden die Eingaben mit Hilfe von GetRawKey verarbeitet. Dieser Befehl liest einzelne Tastendrücke von der Tastatur, ohne auf ein <RETURN> zu warten. Die Eingaben werden durch die Routine auch vorverarbeitet. Bei den mehrere Zeichen langen Codes der F-Tasten zum Beispiel erfolgt eine Umwandlung in Zeichen mit negativem Vorzeichen. Aus diesen Werten wird in der Hauptzeile die eigentliche Kommandozeile generiert. Empfängt die Hauptschleife ein positives Zeichen (also ASCII), fügt Amiga-Shell das Zeichen einfach in die Zeile ein. Bei negativem Vorzeichen werden je nachdem die Routinen zur Cursorpositionierung, Historyverwaltung und Funktionstastenauswertung angesprungen.

Bei einer Funktionstaste wird die Routine »strins()« in Zeile 181 aufgerufen, die den Text der Funktionstaste in die Kommandozeile einfügt. Blättert man mit <Cursor rauf/runter> in der History, wird »printhistory()« (Zeile 380) aufgerufen, was die Kommandozeile durch einen History-Befehl ersetzt.

Bei Drücken von <RETURN>, wird die neue Kommandozeile in die History eingefügt und anschließend »comline()« (Zeile 272) angesprungen. Diese Routine bildet zusammen mit »execbef()« (Zeile 299) das Herz der Amiga-Shell.

In comline() wird die Commandline, falls in ihr mehrere Befehle stehen, zerlegt. Aliasstrings werden durch die entsprechenden Strings ersetzt. Die so erhaltenen Einzelbefehle werden an execbef() übergeben. Diese Routine prüft als erstes, ob es sich um einen Shellinternen Befehl (zum Beispiel SET) handelt. Ist das nicht der Fall, prüft das Programm mit Hilfe des FileInfoBlocks, ob es sich um ein Programm oder ein Directory handelt. Je nachdem wird anschließend das File mit Execute ausgeführt oder das Directory mit cd() angewählt.

Nachfolgend sollen noch einige der Funktionen beschrieben werden.

— stralias():
Verwaltet alias-Anweisungen. Prüft, ob ein entsprechender String schon existiert. Wenn ja, wird er gelöscht, und damit der dynamisch verwaltete Speicher wieder freigegeben. Dann holt sich Amiga-Shell einen Speicherplatz der entsprechenden Größe vom System und speichert den Text der Aliaszuweisung.

— historyin():
Der Speicher für die History wird zyklisch verwaltet. Wenn die aktuelle Position in der Liste (hist) größer als die maximale Anzahl der Einträge (hismax) ist, beginnt die Liste von vorne. Nach dem Löschen des alten History-Eintrags erfolgt der Eintrag des neuen.

— history ():
Legt die Länge der History fest und listet die aktuelle History

— set():
Speichert und löscht die Funktionstastenbelegung

— stralias():
Dieser Programmteil ersetzt die mit ALIAS definierten Texte durch die ihnen zugewiesenen Texte. Da es rekursiv ist, kann ein Aliasstring wieder einen Aliasstring enthalten. Bei passendem Einsatz kann sich das Programm für immer damit beschäftigen (Endlosschleife).

— decode ():
Sieht nach, ob in »text[]« ein Amiga-Shell-Befehl steht, wenn ja, wird die Nummer zurückgegeben, andernfalls 0. Die Befehle müssen in »befehle[]« stehen, und großgeschrieben sein.

— aliasuse():
Ersetzt die aliasnamen durch die zugewiesenen Zeichenketten.

— cd():
Macht das gleiche wie der CLI-Befehl

— input():
Verlangt die Eingabe des zweiten Teils einer Aliaszuweisung durch den Benutzer.

— execfile():
Führt ein Batchfile aus, das auch Amiga-Shell-Befehle enthalten darf.

— histo():
Listet die letzten Eingaben oder legt die Länge der History fest.

— RawModeOn():
Schaltet das Console-Device auf Eingabe von Einzelzeichen um.

— RawModeOff():
Schaltet in den normalen Modus zurück.

— hilfe():
Gibt eine kurze Anleitung aus.

Für diejenigen, die Amiga-Shell noch um eigene Befehle erweitern wollen, ist noch die Variablenliste interessant.

x — Cursorposition in der Kommandozeile
xm — Anzahl der Zeichen in der Kommandozeile
hist — Zeiger auf aktuelles Historyelement für eine neue Zuweisung nach <RETURN>
hismax — maximale Anzahl der Elemente in der History
his — zeigt beim »Blättern« auf das aktuelle Element
strg und text — enthalten die aktuelle Commandline
fkey — Zeiger auf die Funktionstastenstrings
alias — Zeiger auf die Aliasstrings
history — Zeiger auf die Historystrings
befehle — Zeiger auf die Befehlsnamen
end — Flag für Ende (wird durch QUIT gesetzt)
alimax — Anzahl der Aliaszuweisungen
befmax — Anzahl der Befehle

Und falls Sie etwas vergessen haben, erinnert Sie »hilfe« an die Fähigkeiten von Amiga-Shell. Um das Programm zu verlassen, gibt man einfach »quit« ein. Aber wer will schon auf diesen Komfort verzichten? (Dirk Heckmann/rb)

ALIAS [text1] [text2]
EXEC dateiname
CLS
HILFE
QUIT
SET [Funktionstaste] [text]
INPUT name
HISTORY [anzahl]

Tabelle. Die Befehle der »Amiga-Shell«, die Parameter in eckigen Klammern sind optional

Programmname:	Amiga Shell
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40
Aufrufe:	cc amigashell +l, ln amigashell -lc32

```
Programm : Amiga Shell
----------------------

  1 4y0 #include <exec/types.h>
  2 On  #include <libraries/dos.h>
  3 AF  #include <exec/memory.h>
  4 DO  #include <stdio.h>
  5 UF  #include <sgtty.h>
  6 2p  int c,x,xm,in,hist,his,hismax;
  7 d6  char text[200],strg[200];
  8 ùC  char *fkey[24];
  9 uZ  static char *befehle[]={
 10 Ks  "CD","SET","HILFE","QUIT","ALIAS","CLS","INPUT","EXEC","HISTO
        RY",};
 11 zF  char *alias[200];
 12 3u  char *history[100];
 13 ZM  int befmax,ali,alimax,end;
 14 9d  ULONG DosBase;
 15 Jf  main(argc,argv)
 16 gG  int argc;
 17 8F  char *argv[];
 18 Eh  {
 19 Yq1  int i,t,bef,loop,br;
 20 zu   in=1;x=0;xm=0;befmax=9;end=0;hismax=20;
 21 HU   ali=1;alimax=1;
 22 8D   hist=-1;
 23 CJ   /* allle noetigen und unoetigen libraries oeffenen */
 24 jb2   if ((DosBase=OpenLibrary("dos.library",0L))==0)exit();
 25 IL1  /* F-Tasten als undef kennzeichnen */
 26 NV   for (i=0;i<24;i++)
 27 ky2   fkey[i]=0;
 28 j81  /* History auf Null */
 29 Vg   for (i=0;i<90;i++)
 30 7A2   history[i]=0;
 31 zo1  /* alias promt als > setzen */;
 32 jQ   if((alias[0]=AllocMem(8,0))==0) exit();
 33 00   if((alias[1]=AllocMem(2,0))==0) exit();
 34 qu   strcpy(alias[0],"_Prompt");
 35 uF   strcpy(alias[1],"> ");
 36 Sd   RawModeOn();
 37 uQ   /* auf startup sequence checken */
 38 rg   if (argc==2)
 39 Dm2   if (*argv[1]=='?')        /* Hilfe geben */
 40 A73    hilfe();
 41 UH2   else
 42 c5     {
 43 k43     strcpy(&text[0],argv[1]);
 44 gF      execfile(1);    /* startup sequence ausführen */
 45 jE2    }
 46 W30 while (end!=1)
 47 hA  {
 48 q91  x=0;xm=0;
 49 VC   printf("%s",alias[1]);
 50 Gv   text[0]=0;
 51 ai  while((c=GetRawKey())!=13)
 52 mF2  {
 53 nF3   if (c>0)                         /* also ascii */
 54 Ux4    switch(c)
 55 pI     {
 56 Fa5     case 127:if (x<xm) {tdel(x);scr_cdelete();}break;
 57 cw      case 8  :if (x>0) {x--;scr_bs();scr_cdelete();tdel(x);
            }break;
 58 4I      case 9  :if (in==0) in=1;else in=0;break;
 59 fZ      default:
 60 H46      if (in==0||x==xm)
 61 v07       {
 62 AZ8        putchar(c)                /* Zeichen ausgeben */;
 63 gj         text[x++]=c;              /* und in Puffer */;
 64 Nl         xm++;                     /* Puffer ende erhoehen*/
 65 P2         if (xm>100) {xm--;x--;}
 66 4Z7       }
 67 uh6      else
 68 2V7       {
 69 jp8        scr_cinsert();
 70 LE         putchar(c);
 71 Hk         binsert(x);
 72 eU         text[x++]=c;
 73 xE         xm++;
 74 Ch7       }
 75 Di4    }
 76 3q3   else
 77 yi4    if (c<-20)
 78 sL6     switch(c)
 79 Dg7      {
 80 3q8       case -30:printhistory(-1);break;
 81 It        case -31:printhistory( 1);break;
 82 br        case -32:if (x<xm) {x++;scr_cursrt();}break;
 83 Xl        case -33:if (x>0)  {x--;scr_bs()    ;}break;
 84 Mr7      }
 85 8U5    else  /* F-Key auswerten */
 86 Qs6     if (fkey[-c]!=0)
 87 8d7      strins(fkey[-c]);/* fkey in text einfuegen */
 88 Qv2  }
 89 8o1 text [xm]='\0';
 90 Ve  putchar('\n');
 91 Qc  historyin();
 92 qj  delspace();  /* space vor erstem zeichen */
 93 63  if (strlen(&text[0])!=0) comline();
 94 W10 }
 95 Uq1 /* Programm verlassen und Mem freigeben */
 96 nj  /* alias freigeben */
 97 pt  for (i=0;i<(alimax*2);i++)
 98 ON2  if (alias[i]!=0)
 99 sd3   FreeMem(alias[i],strlen(alias[i])+1);
100 za1 /* Speicher fuer di F-Tasten */
101 OT  for (i=0;i<21;i++)
102 BF2  if (fkey[i]!=0)
103 Kw3   FreeMem(fkey[i],strlen(fkey[i])+1);
```

```
104 lJ1  /* History freigeben */
105 OS   for (i=0;i<20;i++)
106 uu2   if (history[i]!=0)
107 L73    FreeMem(history[i],strlen(history[i])+1);
108 up1  /* normal Consolen Modus */
109 p0   RawModeOff();
110 mH0 }
111 35  binsert(x)
112 Bj  int x ;
113 lE  {
114 aG1  int v;
115 Dd   for (v=160;v>=x;v--)
116 tW2   text[v+1]=text[v];
117 t00 }
118 XN  tdel(x)
119 rP  int x;
120 sL  {
121 Rx1  int i;
122 AR   for (i=x;i<160;i++)
123 ja2   text[i]=text[i+1];
124 my1  if(xm>0) xm--;
125 1W0 }
126 e3  sdel()
127 zS  {
128 Y41  int i;
129 Yr   for (i=0;i<199;i++)
130 e82   strg[i]=strg[i+1];
131 7c0 }
132 Ji  delspace()
133 5Y  {
134 nR1  while(text[0]==32)
135 EL2   tdel(0);
136 Ch0 }
137 U3  decode ()
138 Ad  {
139 Tg1  int len,bef,r,befehl,i,sd;
140 ql   char *string;
141 r1   befehl=0;
142 s8   for (bef=0;bef<befmax;bef++)
143 Fi2   {
144 UH3    string=befehle[bef];
145 b5     r=0;
146 KR     len=strlen(string);
147 od     for (i=0;i<len;i++)
148 HI4     if (toupper(text[i])!=*(string++)) r=1;
149 K23    if (text[len]!=0 && text[len]!=32) sd=1;else sd=0;
150 aC     if (r==0 && sd==0) befehl=bef+1;
151 Rw2   }
152 6v1  return(befehl);
153 Ty0 }
154 bW  setfkey()
155 Ru  {
156 ZU1  int fk;
157 rD   fk=-1;
158 hk   tdel(0);tdel(0);tdel(0); /* set loeschen */
159 Gp   delspace();               /* spacezeichen bis zum f loeschen
     */
160 i8   if (text[0]==0)
161 X02   {
162 dY3    for (fk=1;fk<21;fk++)
163 RL4     if (fkey[fk]!=0)
164 WQ6      printf("Key %2d   %s\n",fk,fkey[fk]);
165 Et3    return ();
166 gB2   }
167 SV1  if (text[0]=='f') fk=0 ;
168 Sq   if (text[0]=='F') fk=10;
169 c9   if (fk==-1) {printf("Def siehe Handbuch");return();}
170 U0   fk=fk+text[1]-'0';
171 IC   if (fk==10) fk=20;
172 eY   if (fk==0 ) fk=10;
173 iN   if (fk<1 || fk>20) {printf("F-Key S 22");return();}
174 7M   tdel(0);tdel(0);tdel(0);/* fx_ loeschen */
175 dX   if (fkey[fk]!=0)
176 zn2   FreeMem(fkey[fk],strlen(fkey[fk])+));
177 OA1  fkey[fk]=AllocMem(strlen(&text[0])+1,0);
178 ga   if (fkey[fk]!=0)
179 Cr2   strcpy(fkey[fk],&text[0]);
180 uP0 }
181 xi  strins(string)
182 WR  char *string;
183 tM  {
184 JU1  int t,i;
185 oW   t=strlen(string);
186 dV   for (i=0;i<t;i++)
187 xQ4   {
188 Ad5    binsert(x);
189 BG     text[x]=*string;
190 gm     scr_cinsert();
191 uY     putchar(text[x]);
192 kA     x++;string++;xm++;
193 7c4   }
194 8d0 }
195 d9  aliasuse()
196 6Z  {
197 VD1  int flag,i,t,f,len,pos,wimpel ;
198 fg   char buff[100],c;
199 nG   len=0;
200 2C   while (text[len++]!=0); /* zeilen laenge =bef laenge */
201 Fe   len--;
202 qE   i=-1;
203 At   for (t=0;t<alimax;t++)   /* alias finden */
204 sP2   if (!strncmp(alias[2*t],&text[0],len)) i=t;
205 xI1  if (i!=-1) strcpy(&text[0],alias[2*i+1]);
206 09   /* inlin alias erkennen */
207 qt0 wimpel=0;
208 Fw  while(wimpel==0)
209 Jm  {
210 x11  wimpel=1;
211 bi   len=strlen(&text[0]);
212 Mv   t=-1;
213 fT   for (i=0;i<len-1;i++) /* alias suchen */
214 Or2   {
215 He3    if (text[i]=='')
216 r54     {t=i;}
217 V02   }
218 c01  pos=t;
219 SR   if (t!=-1)  /* alias gefunden */
220 Ux2   {
221 473    wimpel=0;
222 Yt     i=0;
223 3d     tdel(pos);
224 6T     flag=0;
225 U3     while (flag==0)
226 a34     {
227 ff5      buff[i]=text[t];
228 zH       if (buff[i]==32 || buff[i]==0) flag=1;
229 EZ       i++;t++;
230 iD4     }
231 9A3    buff [i]=0;len=i;
232 wO     for (i=0;i<len-1;i++) /* zuweisung loeschen */
233 Dn4     tdel(pos);
234 gU3    if (text[pos]==32) tdel(pos);
235 Es     xm=xm-len;
236 Et     /* Zeiger auf alias finden */
237 Je     f=-1;
238 9c     for (i=0;i<alimax;i++)
239 Gp4     if (!strncmp(&buff[0],alias[i*2],len-1)) f=i;
240 Yr3    if (f!=-1)
241 hr4     { /* alias einfuegen */
242 c05      for (i=0;i<strlen(alias[f*2+1]);i++)
243 rK6       {
244 BU7        binsert(pos);
245 08         text[pos++]=*(alias[f*2+1]+i);
246 k1         xm++;
247 zU6       }
248 0V4     }
249 1W2   }
250 2X0 }
251 3Y  }
252 21  cd ()
253 1U  {
254 6f1  BPTR dir,l1;
255 Xk   struct FileInfoBlock *eintrag;
256 Rz   eintrag=(struct FileInfoBlock *)
257 Mh5      AllocMem(sizeof(struct FileInfoBlock),MEMF_CLEAR);
258 1g1  tdel(0);tdel(0);
259 E4   delspace();
260 1Y   dir=Lock(&text[0],ACCESS_READ);
261 ZV   if (dir==NULL)
262 Ad2   {
263 bH3    printf("Directory &s nicht vorhanden\n",&text[0]);
264 Tf     FreeMem(eintrag,sizeof(struct FileInfoBlock));
```

Listing. Komfort im CLI mit »Amiga-Shell«. Listing bitte ohne Zeilennummern eingeben. Bitte den Checksummer auf Seite 76 verwenden.

```
265 8y     return();
266 In2   }
267 Hp1  l1=CurrentDir(dir);
268 Kn   UnLock(l1);
269 dc   Examine(dir,eintrag);
270 Z1   FreeMem(eintrag,sizeof(struct FileInfoBlock));
271 NsO }
272 dO  comline()
273 Lo  {
274 Ce1  int bef,i,c,r,flg;
275 Vi   bef=decode();
276 B7   if (bef!=2 && bef!=5 && bef!=8)    * alias und set */
277 Ps   {
278 qn2   aliasuse();
279 fP    strcpy(&strg[0]&&text[0]);
280 GS    while (strg[0]!=0)
281 Tw    {
282 oI3    r=0;
283 BN     while (strg[0]!=';' && strg[0]!=0)
284 Wz     {
285 3M4     text[r]=strg[0];
286 gn      r++;
287 YO      sdel();
288 e93    }
289 Fy     text[r]=0;
290 my     if (strg[0]==';') sdel();
291 ka     delspace();
292 OZ     if (text[0]!=0 && text[0]!=13)
293 4A4     exebef();
294 kF2   }
295 1G1  }
296 bO   else
297 8E2   exebef();
298 oJO }
299 ze  exebef()
300 mF  {
311 zc1  int bef,i,r,flag;
302 vl   delspace();
303 xA   bef=decode();
304 IH   if (bef!=0)  /* befehl erkennen und ausfuehren */
305 NQ2   switch(bef)
306 sL    {
307 bE3    case 1: cd()        ;break;
308 93     case 2: setfkey()   ;break;
309 hr     case 3: hilfe()     ;break;
310 sV     case 4: end=1       ;break;
311 SI     case 5: stralias()  ;break;
312 T4     case 6: scr_clear() ;break;
313 Hk     case 7: input()     ;break;
314 QC     case 8: bef=execfile(0) ;break;
315 XI     case 9: histo()     ;break;
316 6b2   }
317 sw1  else  /* dos befehl */
318 4X   {
319 rR2   /* space im text => kein cd */
320 qx    i=0;flag=0;
321 Cy    while(text[i]!=0)
322 8b    {
323 A03    if (text[i]==32) flag=1;
324 Oy     i++;
325 Fk2   }
326 6R    if (flag == 0)
327 TY3    if (checkdir()==1)
328 V34    {                       /* atomatisch cd */
329 dw5     binsert(0);
330 x7      binsert(0);  /* weil cd 2 Zeichen loescht */
331 eM      cd();
332 Mr4    }
333 Cz3    else
334 OJ     Execute(&text[0],OL,OL);
335 UD2   else                     /* File usfuehren * */
336 213    Execute(&text[0],OL,OL);
337 Rw1  }
338 SxO }
339 1V  checkdir()
340 Qt  {
341 VV1  BPTR dir;
342 Vo   struct FileInfoBlock *fib;
343 tp   int r;
344 qR   /* ram fuer fib */
345 yO   fib=AllocMem(sizeof(struct FileInfoBlock),MEMF_CLEAR);
346 9w   dir=Lock(&text[0],ACCESS_READ);
347 rL   r=0;
348 On   if (dir==0) r=0;
```

```
349 SF   else
350 hh2   if (Examine(dir,fib)!=0)
351 oU3    if(fib->fib_DirEntryType>0) r=1;
352 T91  UnLock(dir);
353 mO   FreeMem(fib,sizeof(struct FileInfoBlock));
354 RQ   return(r);
355 JEO }
356 5E  input ()          /* eingabe eines alias */
357 hA  {
358 Q71  int len;
359 iq   char c;
360 07   len=strlen(&text[0]);
361 Y1   text[len]=32;      /* space namen einfuegen */
362 u5   RawModeOff();
363 Op   while((c=getchar())!='\n')
364 rv3    text[++len]=c;
365 2I1  text[len+1]=0;
366 mx   RawModeOn();
367 nV   stralias();
368 wRO }
369 bM  historyin()
370 uN  {
371 8J1  int i,r,t,flag;
372 ht   hist=hist+1;
373 ra   if (hist>19) hist=0;
374 kd   his=hist+1;
375 he   if (history[hist]!=0)
376 lk2   FreeMem(history[hist],strlen(history[hist])+1);
377 Hc1  history[hist]=AllocMem(strlen(&text[0])+1,MEMF_CLEAR);
378 8m   strcpy(history[hist],&text[0]);
379 7cO }
380 c1  printhistory(dr)
381 EO  int dr;
382 6Z  {
383 n51  int t,i,r,flag;
384 5W   his=his+dr;
385 p2   if (his>hismax) his=0;
386 HV   if (his<0 ) iis=hismax;
387 2C   if (history[his]!=0)
388 Cf   {
389 av2   putchar(13); /* an anfang der Zeile */
390 Xu    printf("
                 ");
391 B1    putchar(13);
392 2j    printf("%s",alias[1]);
393 1y    printf("%s",history[his]);
394 Zo    strcpy(&text[0],history[his]);
395 7Z    xm=strlen(history[his]);
396 8w    x=xm;
397 Pu1  }
398 F2   else
399 Nq   {
400 Ku2   putchar(13);
401 i5    printf("
                 ");
402 Mw    putchar(13);
403 Du    printf("%s",alias[1]);
404 JX   xxm=0;x=0;
405 X21  }
406 Y30 }
407 UO  execfile(flag)
408 GO  int flag;
409 XO  {
410 Xv1  int datei,i,t,r;
411 Yg   char c;
412 HT   FILE *fp,*fopen();
413 d3   c=15;
414 rv   if (flag==0)        /* normaler aufruf */
415 Uu   {tdel(0);tdel(0);tdel(0);tdel(0);}
416 lb   delspace();
417 zX   fp=fopen(&text[0],"r");
418 vk   if (fp==0)
419 ts2   printf("File %s nicht gefunden\n",&text[0]);
420 bO1  else
421 qp2   while (c!=EOF)
422 kD    {
423 n83    i=0;
424 xO     while ((c=getc(fp))!=EOF &&  c!=0x0a)
425 rS4     text[i++]=c;
426 yE3    text[i]=0; /* end of line */
427 wm     delspace();
428 WE     if (text[0]!=0 && text[0]!='#' && text[0]!=';')
           comline();
429 v2    }
```

```
430 4w1  fclose(fp);
431 P4   text[0]=0;
432 S8   text[1]=0;
433 L1   return(30);
434 OVO }
435 nJ  hilfe()
436 yR  {
437 fs1  printf(" by Dirk Heckmann 15.09.1987\n\n");
438 xH   printf("mit 'set fx $' kann man dir F-Tasten belegen.\n");
439 M5   printf("x ist eine Zahl zwischen 0 und 9 und gibt die Taste
         an.\n");
440 Mu   printf("0=10 1=1 2=2 ... 9=9 . Mit SHIFT F kann man die ge-\
         n");
441 eJ   printf("shifteten F-Tasten definieren. $ ist der gewünschte
         string\n\n");
442 TH   printf("Mit  'alias $0 $1' kann man String $0 die $1 zuweise
         n.\n");
443 yU   printf("Gibt man dann $0 ein wird zum Ausführen $1 verwendet
         .\n");
444 3n   printf("Will man innerhalb eines Textes einen alias $ einfüg
         en\n");
445 18   printf("muss man in mit  (alt i) davor und SPACE danach sch
         reiben.\n");
446 p5   printf("2 Space wenn hinter der $ wenn noch 1 Space bleiben
         soll.\n\n");
447 pv   printf("Mit 'exec filename' kann man ein BatchFile auf die i
         nternen\n");
448 f0   printf("Befehle anwenden (alias,set).\n\n");
449 IJ   printf("Der cd Befehl braucht nicht mehr eingegeben zuwerden
         ,\n");
450 sG   printf("das Prog erkennt selbstständig ob es sich um ein DIR
          \n");
451 7k   printf("oder ein File handelt.\n");
452 pw   printf("Input funktioniert wie alias, nur das der Benutzer $
         1 ein-\n");
453 G1   printf("geben muss (wie Input bei Basic).\n");
454 KpO }
455 fs  histo()
456 I1  {
457 eB1  int i,flag;
458 4x   for (i=0;i<7;i++)
459 CW2   tdel(0);              /* 'history' loeschen' */
460 TJ1  delspace();
461 ua   flag=0;i=hist;
462 a0   if (text[0]==0)
463 Kt2   while (flag==0)
464 Qt     {
465 HF3     i++;
466 ji      if (i>hismax) i=0;
467 Qz      if (i==hist) flag=1;
468 kk      if (history[i]!=0)
469 Q84      printf("%s\n",history[i]);
470 a52    }
471 QD1  else
472 Y12   {
473 KA3    if ((flag=text[0]-48)<0 || flag>9) return();
474 ho     tdel(0);
475 RP     if (text[0]!=0)
476 c5      {
477 Fn4      flag=flag*10;
478 cn       if ((i=text[0]-48)<0 || i>9) return();
479 y6       flag=flag+i;
480 kF3     }
481 tC     if (flag>2) hismax=flag;
482 N4     printf("new History %d Lines\nOld History destroyed\n",his
         max);
483 KE     his=1;hist=0;
484 oJ2   }
485 pKO }
486 q7  stralias()
487 nG  {
488 AW1  int i,len,el,k,t,flag;
489 8d   char name[100];
490 UL   for (i=0;i<5;i++)
491 y52   tdel(0);
492 zp1  delspace();
493 ui   if (text[0]==0)  /* alias liste */
494 uN    {
495 I12    for (i=0;i<alimax;i++)
496 8t3     printf("%s  %s\n",alias[2*i],alias[2*i+1]);
497 si2    return();
```

Listing. Komfort im CLI mit »Amiga-Shell« (Fortsetzung)

```
498 2X1  }
499 jc   len=0; /* naechstes space suchen */
500 RZ   while (text[len]!=0 && text[len]!=32) len++;
501 1g   /* element finden */
502 Rn   el=-1;
503 Qt   for (i=0;i<alimax;i++)
504 iL2   if (!strncmp(&text[0],alias[2*i],len) && len==strlen(alias[
          2*i]))
505 Qf3    el=i;
506 Ym1  /* namem merken */
507 cR   for (i=0;i<len;i++)
508 6t2   name[i]=text[i];
509 Uk    name[len]=0;
510 Q41  /* und aus text loeschen */
511 gV   for (i=0;i<len;i++)
512 JQ2   tdel(0);
513 KA1  delspace();
514 mw   if (el==0 && text[0]==0) return(); /* _Prompt sollte bleiben
          */
515 UC   /* altes alias loeschen */
516 kh   if (el!=-1)
517 Hk2   {
518 Aa3    FreeMem(alias[2*el  ],strlen(alias[2*el  ])+1);
519 KO     FreeMem(alias[2*el+1],strlen(alias[2*el+1])+1);
520 Ot2   }
521 mT1  /* ggf liste verschieben */
522 Bp   if (text[0]==0 && el!=-1)   / element aus alias loeschen */

523 Nq    {
524 fF2    for (i=el;i<alimax;i++)
525 Ps     {
526 763     alias[2*i  ]=alias[2*i+2];
527 pO      alias[2*i+1]=alias[2*i+3];
528 W12    }
529 Om     alimax--;
530 PF     return();
531 Z41   }
532 fC   if (el==-1 && alimax>98) return(); /* alles voll */
533 Oa   if (el==-1) el=alimax++;
534 Az   /* Element neue schreiben */
535 2y   alias[2*el  ]=AllocMem(strlen(&name[0])+1,MEMF_CLEAR);
536 7s   alias[2*el+1]=AllocMem(strlen(&text[0])+1,MEMF_CLEAR);
537 o3   strcpy(alias[2*el  ],&name[0]);
538 AY   strcpy(alias[2*el+1],&text[0]);
539 hCO }
540 uI  RawModeOn ()
541 f8  {
542 eY1  int c;
543 qR   FILE *fp;
544 4d   struct sgttyb stty;
545 QQ   /* RAW Mode einschalten */
546 ZL   fp=stdin;
547 Au   ioctl(fileno(fp),TIOCGETP,&stty);
548 QM   stty.sg_flags| = RAW;
549 mi   ioctl(fileno(fp),TIOCSETP,&stty);
550 sNO }
551 UE  RawModeOff ()
552 qJ  {
553 Ob1  FILE *fp;
554 En   struct sgttyb stty;
555 iU   fp=stdin;
556 J3   ioctl(fileno(fp),TIOCGETP,&stty);
557 LS   stty.sg_flags=0; /* Default Mode */
558 vr   octl(fileno(fp),TIOCSETP,&stty);
559 1WO }
560 aD  GetRawKey ()
561 zS  {
562 AN1  int c1,c2,c,cb;
563 ko   c=getchar();
564 LI   cb=c;
565 Rt   if (c==155)
56 4X2   {
567 HP3    cb=0;
568 pt     c=getchar();
569 nG     switch(c)
570 8b     {
571 1W4     case 68:cb--;                        /* Cursor Tasten */
572 Xw      case 67:cb--;
573 Vt      case 66:cb--;
574 qH      case 65:cb=cb-30;break;
575 Nq      case 63:cb=-22;c=getchar();break;     /* HELP Taste */
576 X3      case 84:cb=-35;break;                /* up + SHIFT */
577 tB      case 85:cb=-36;break;
578 Kp3    }
579 6A     if (c==32)
580 I15      {
581 267       c=getchar();
582 Ve        if (c==64)
583 Zy9        cb=-37;
584 F28       else
585 d39        cb=-38;
586 Sx6      }
587 uh3    if (cb==0)
588 Qt4     {
589 yY5      c1=getchar();
590 7b       if (c1!=126)
591 Tw6       {
592 Xp7        cb=-10        /* F-Tasten mit SHIFT */ ;
593 Wk         c=getchar()   /* F-Tasten end Code lesen */;
594 Sc         c=c1;
595 b66       }
596 it5      cb=cb-c+47;
597 d84     }
598 e92   }
599 9P    return(cb);
600 gBO }
(C) 1987 M&T
```

Listing. Komfort im CLI mit »Amiga-Shell« (Schluß)

Fortsetzung von Seite 42

beide Werte auf die gewünschte Zahl ein.

Die größtmögliche Flexibilität erhält man nun noch durch die Auswahl der Diskettenseite. Mit »Head« läßt sich bestimmen, welche Seite kopiert werden soll. Dabei bedeuten:

```
0 - Nur Seite 0
1 - Nur Seite 1
2 - Beide Seiten
```

Die Angabe, wo sich die Quelldiskette befindet, erfolgt beim Menüpunkt »Source«. Die Anzahl der benutzbaren Ziellaufwerke ist abhängig von der Zahl angeschlossener Laufwerke. Maximal werden jedoch vier Laufwerke verwaltet. Die Ziellaufwerke 2 und 3 können durch die Eingabe von <n> abgeschaltet werden. Zum Einschalten gibt man einfach die Nummer des Laufwerks ein. Werden für Quell- und Ziellaufwerk dieselbe Zahl eingegeben, schaltet das Programm automatisch alle anderen Ziellaufwerke ab.

Mit <ESC> kann nach erfolgter Einstellung das Untermenü verlassen werden. Geschieht nach dem Drücken von <ESC> nichts, wurde ein Parameter falsch gewählt.

Der Start des Kopiervorgangs erfolgt mit <F10>. Nun zeigt DCopy, welche Diskette wo einzulegen ist und wartet auf das Drücken von <RETURN>. Wird mit nur einem Laufwerk gearbeitet, berechnet DCopy die nötige Anzahl der Diskettenwechsel und gibt sie auf dem Bildschirm aus.

Beim Kopieren selbst wird angezeigt, welche Spur DCopy gerade liest oder schreibt. Lese- und Schreibfehler während des Kopierens zeigt das Programm an. Kann die Diskette nicht kopiert werden (Abbruch oder Verify-Error), erfolgt eine Anzeige im Fenster.

Um DCopy zu verlassen, genügt ein Druck auf <ESC>. Vorher gibt DCopy jedoch noch die Anweisung, sämtliche Kopien aus den Laufwerken zu entfernen. Dies ist nötig, da die Kopien identisch zum Original sind und Amiga-DOS Probleme bekäme. Genau dasselbe tritt auf, wenn sich vor dem Start von DCopy zwei exakt gleiche Disketten in den Laufwerken befinden. In so einem Fall müssen Sie erst das Programm starten und dann die Disketten einlegen.

Nach der Beendigung von DCopy können Sie selbstverständlich ganz normal mit Ihrem Amiga weiterarbeiten. Und das viel ruhiger, da Sie ja dann Sicherheitskopien von Ihren Programmen und Daten besitzen. (Gerald Fries/rb)

Joystickabfrage

Wer den Joystick für ein Programm benutzen will, kann aus dem »Amiga Reference Manual« die Routine abtippen. Im Gegensatz dazu ist unser Listing »Joy« sehr kurz und kann ohne Probleme in jedes C-Programm eingebunden werden.

Für Spiele benötigt man immer wieder den Joystick, da manche Steuerung damit genauer ist als mit der Maus. Das oben schon erwähnte C-Programm aus dem Reference Manual ist vier Seiten lang und nicht gerade einfach zu verstehen. Man kann sich die Werte für die zwei Joysticks auch direkt aus den Registern holen.

Zu diesem Zweck muß man allerdings erst einmal wissen, in welchen Registern welche Werte stehen. Für das Programm »Joy« (siehe Listing 1) benötigen wir drei Register. Das erste liegt bei Adresse $bfe001 und enthält in den Bits 6 und 7 die Information über den Feuerknopf der Joysticks in Port 1 und 2. Die Adressen $dff00a (Port 1) und $dff00c (Port 2) enthalten die nötigen Werte für die Richtung (siehe Bild).

Nun müssen wir noch die erhaltenen Werte richtig miteinander verknüpfen. Für die Richtungen »links« und »rechts« ist dies sehr einfach. Dazu müssen wir nur die Bits Y1 beziehungsweise X1 testen. Erhalten wir den Wert 0, wird nicht in diese Richtung gedrückt, bei dem Wert 1 wissen wir die Richtung.

Etwas komplizierter wird es mit »vorwärts« und »rückwärts«. Ob zum Beispiel nach vorne gedrückt wird, erfahren wir, indem wir eine Exklusiv-Oder-Verknüpfung mit dem nächst niedrigeren Bit durchführen. Für vorwärts müssen wir also die Bits Y1 und Y0 Exklusiv-Oder verknüpfen. Erhalten wir dann den Wert 1, wurde der Joystick nach vorne bewegt. Für die entgegengesetzte Richtung wird analog verfahren, nur werden hier X1 und X0 benutzt. Ob der Feuerknopf gedrückt wurde, erfahren wir aus dem Register $bfe001. Diesen Wert setzen wir später in unser Ergebnis im Bit X4 ein, um es im Hauptprogramm auszuwerten.

Anhand der folgenden Erläuterung des Programms »Joy« wird dies noch klarer ersichtlich.

Zeile 5 und 6:
Die Funktion »joystick« verlangt einen Parameter vom Typ short (16 Bit).

Zeile 8 bis 11:
Folgende Variablen werden für die Routine verwendet.
wert — Ein Zeiger auf die jeweilige Adresse des Joystickports
cia — Ein Zeiger auf das Register ($bfe001) in dem die Feuerknopfwerte stehen
erg — Das Ergebnis der Routine

Zeile 14:
Die Adresse des Joystickports wird berechnet, indem zur Adresse $dff008 der übergebene Parameter (port, Werte 1 oder 2) verdoppelt und addiert wird.

Zeile 16:
Der Inhalt der Adresse wird in erg übernommen

Zeile 20 bis 23:
Mit der if-Abfrage wird entschieden, ob das Bit für den Feuerknopf gesetzt oder gelöscht wird. Der Ausdruck »64 * port« ergibt dabei die Werte 64 oder 128, also Bit 6 oder 7. Durch die Und-Verknüpfung mit dem Inhalt von »cia« erhält man den Wert 1, falls der Knopf gedrückt ist.

Zeile 21 und 23:
Hier wird durch eine Und- beziehungsweise Oder-Verknüpfung das Bit X4 im Ergebniswert erg gelöscht oder gesetzt

Zeile 25:
Das Ergebnis wird an das aufrufende Programm übergeben

Das Demoprogramm »Demojoy« (siehe Listing 2) dient nur zur Vorführung der Funktion »joystick«. Auch hier wird eine Variable vom Typ short (»wert«) definiert. In ihr befindet sich später das Ergebnis der Funktion. In Zeile 13 wird die Routine joystick aufgerufen und der Rückgabewert in wert gespeichert. Hier wird der Joystick in Port 2 getestet. Wenn Sie den anderen Port auslesen wollen, ersetzen Sie die »2« durch eine »1«.

Nun folgen fünf if-Abfragen, in denen dann der entsprechende Text für die jeweilige Richtung ausgegeben wird. Interessant ist dabei die Berechnung, die für die Exklusiv-Oder-Verknüpfung nötig ist. In Zeile 18 haben Sie zum Beispiel die Berechnung

```
((wert >> 1) ^ wert) & 256
```

Zuerst wird der 16 Bit Wert um eine Stelle nach rechts verschoben, dies geschieht mit

```
(wert >>1)
```

Danach erfolgt die Eklusiv-Oder-Verknüpfung mit dem ursprünglichen Wert. Da der erste Wert verschoben wurde, liegen nun Y0 und Y1 genau übereinander. Mit der Und-Verknüpfung (» & 256«) erhält man nun den Wert der gesamten Verknüpfung. Ist dieser 1, ist der Feuerknopf gedrückt.

Wie Sie in Zeile 2 noch sehen können, kann man die Funktion einfach mit einer » # include«-Anweisung einbinden, da sie keine anderen Funktionen benötigt. (Andreas Reiser/rb)

Sie müssen nur das Programm demojoy übersetzen, da die Unterroutine mit einer include-Anweisung eingebunden wird.

Aztec-C:

```
cc demojoy +1
ln demojoy.o -1c32
```

Lattice-C:

```
lc demojoy
blink lib:c.o,demojoy.o lib lib:lc lib,lib:amiga.lib
```

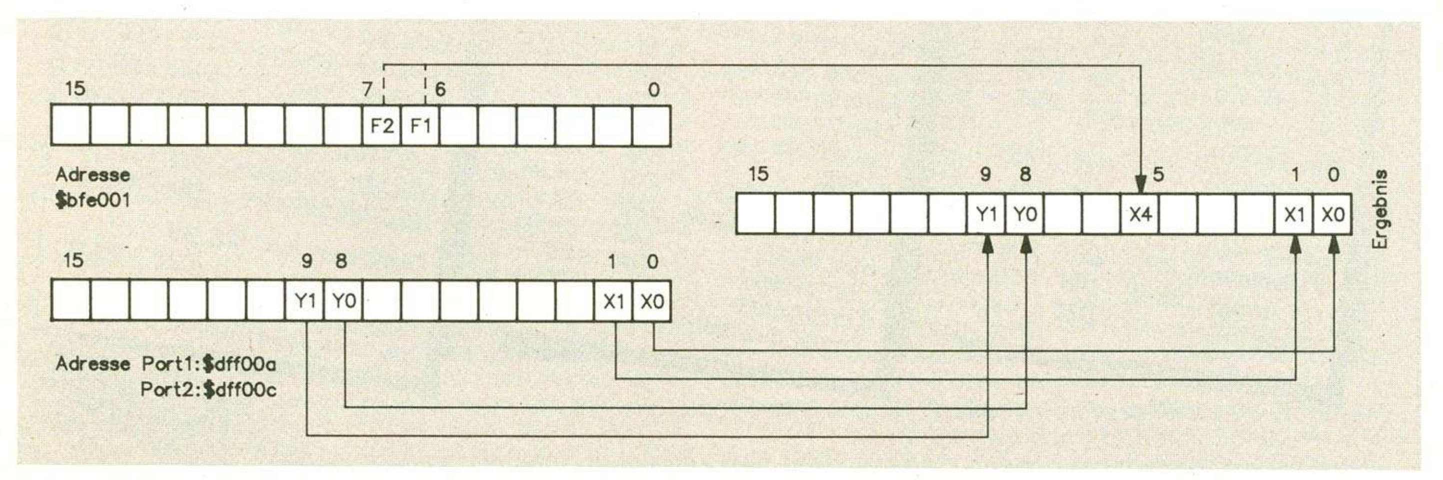


Bild. Aus diesen Registern werden die benötigten Werte für die Joystick-Abfrage geholt und in das Ergebniswort eingetragen

```
 1 C00 /*******************************/
 2 1T  /*      Joystick-Routine       */
 3 S8  /* Autor: Andreas Reiser       */
 4 F3  /*******************************/
 5 6N  short joystick(port) /* Unterpgm-name */
 6 gG  short port;
 7 3W  {
 8 kb1  short *wert;
 9 7d   short erg;
10 NR0 /* Adresse fuer Feuerknopfabfrage */
11 261  char *cia = 0xbfe001;
12 t00 /* Joystickadresse des jeweiligen */
13 2I  /* Ports (1 od. 2) wird ermittelt */
14 us1  wert = (0xdff008 + 2 * port);
15 xm0 /* Inhalt der Adresse wird gelesen */
16 SL1  erg = *wert;
17 Ld0 /* Wenn Feuerknopf gedrueckt ist,  */
18 Be  /* wird Bit5 auf 1 gesetzt, andern-*/
19 CH  /* falls wird Bit5 auf 0 gesetzt.  */
20 5g1  if (*cia & (64 * port))
21 pY4     erg = (erg & 0xffef);
22 By1  else
23 xc4     erg = (erg | 0x0010);
24 jz0 /* Ruecksprung ins Hauptprogramm    */
25 2j1  return(erg);
26 Qv0 }
(C) 1987 M&T
```

Listing 1. »Joy«, eine kurze Funktion zur Abfrage des Joysticks. Bitte ohne Zeilennummern eingeben.

```
 1 AL0 #include <stdio.h>
 2 Bu  #include <joy.c>
 3 SG  /************************************/
 4 ZU  /* Demoprogramm zur Joystick-Routine */
 5 UI  /************************************/
 6 n9  main()
 7 3W  {
 8 071  short joystick();
 9 Qx   short wert;
10 OF   /* Endlosschleife */
11 7o   for(;;)
12 8b   {
13 7c4     wert = joystick(2);
14 gL      if (wert & 2)
15 zK7        printf("rechts ");
16 hF4     if (wert & 512)
17 fW7        printf("links  ");
18 fv4     if (((wert >> 1) ^ wert) & 256)
19 Wu7        printf("vorwaerts ");
20 FZ4     if (((wert >> 1) ^ wert) & 1)
21 nu7        printf("rueckwaerts ");
22 KK4     if (wert & 16)
23 Qi7        printf("feuer");
24 Tw4     printf("\n");
25 Pu1  }
26 Qv0 }
(C) 1987 M&T
```

Listing 2. »Demojoy« dient zur Vorführung von »joy« und gibt die ermittelten Werte auf dem Bildschirm aus

Programmname:	Joy
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.20 oder Lattice-C V3.10
Aufrufe:	siehe Text

Programmname:	Demojoy
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.20 oder Lattice-C V3.10
Aufrufe:	siehe Text

Die Public Domain-Seite

Der Public Domain-Pool wächst! Inzwischen existieren etwa 300 Disketten in verschiedenen Public Domain-Seiten. Immer mehr Programme werden als Freesoft zu günstigen Preisen angeboten. Welche Programme und Disketten empfehlenswert sind, erzählt Ihnen diese Seite.

Jeden Monat tauchen neue Sterne am Public Domain-Markt auf. Seien es Freesoft-Serien wie »TBAG«, »ACS«, »Tornado«, »Panorama« oder die inzwischen 102 Disketten umfassende »Fred Fish«-Sammlung. Insgesamt macht das über 300 Disketten. Leider sind nicht alle PD-Disketten »einzigartig«, da viele Updates auf späteren Disketten erschienen sind und auch viele Programme mehrfach vorhanden sind. Wir stellen in dieser Serie lohnenswerte PD-Disketten vor, von der wir meinen, daß sie in jede Sammlung gehören.

Wer Gags mag und sich gerne kleine Tricks ansieht, wird von der »Tornado 3« begeistert sein. Verschiedene Programmierer haben im Laufe der Zeit kleine Routinen geschrieben, die die seltsamsten Dinge mit dem gerade sichtbaren Bildschirminhalt anstellen. Ein Programm beispielsweise läßt den Bildschirminhalt wie schmelzendes Eis zerfließen, ein anderes macht die Fenster für den Mauszeiger unerreichbar. Insgesamt enthält die Diskette etwa ein Dutzend dieser sich nicht verbreitenden und daher gutartigen »Viren« und ist sicher der Effekte wegen ein lohnender Kauf.

Eine Sammlung hilfreicher Routinen

Nützliche Tools und Hilfsprogramme finden sich auf »Tornado 5«. Diese Diskette enthält einen Zeichensatz- und einen Spriteeditor, das kleine Telefonverzeichnis »BlackBook« und ein Programm, mit dessen Hilfe mit dem Blitter herumexperimentiert werden kann. Auch an Sternforscher wurde gedacht: »StarChart« bringt Sternbilder auf den Bildschirm, wobei durch Anklicken verschiedener Sterne nähere Informationen erscheinen. Besonderes Augenmerk verdient das Programm »NewGetImage«, mit dem beispielsweise Brushes in VSprites oder Bobs umgewandelt werden können. Diese nützliche Diskette ist es wert, in Ihre PD-Sammlung aufgenommen zu werden.

Verschiedene Fractalprogramme erhält der Käufer der »Tornado 8«. Zum einen das Programm »Landscaped«, das

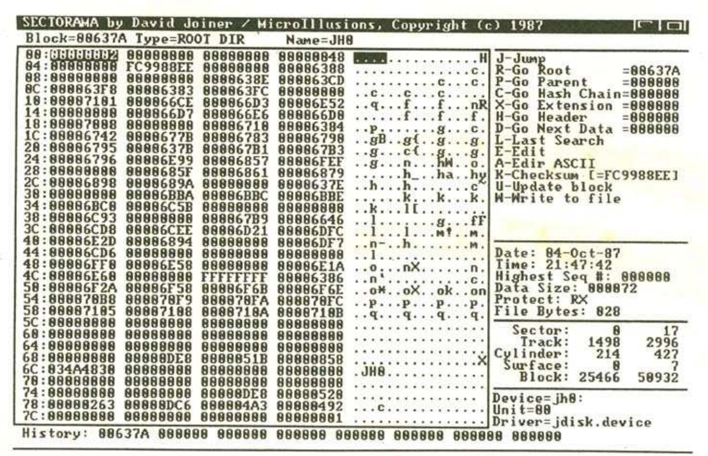


Bild. Der Diskettenmonitor »Sectorama« erlaubt auch Zugriffe auf (fast) beliebig große Festplatten

fractale Landschaften (Meer und Berge mit Schneekuppen) erstellt und zum anderen den »FractalGenerator«, der Bilder erzeugt, die aussehen wie Wärmediagramme. Außerdem findet man auf dieser Diskette neue Versionen der Apfelmännchen-Routinen »MandFXP« und »IMandelVroom«, beides Programme, die schöne Mandelbrot-Mengen berechnen. Oder haben Sie Interesse an einer dreidimensionalen, rotierenden Weltkugel? »3DWorld« macht's möglich. Als nette Dreingabe ist noch das fractale Ballerspiel »Triclops Invasion« enthalten, das diese wirklich gute PD-Diskette angenehm abrundet.

Möchten Sie gerne wissen, wie schnell Ihre Diskettenlaufwerke oder Festplatten tatsächlich im Vergleich zu anderen sind? »DiskPerfA« von Fish »48« hilft Ihnen da weiter. Es gibt Auskunft darüber, wie viele Bytes pro Sekunde gelesen und geschrieben oder wie schnell Directory-Einträge gefunden werden. Da außerdem noch eine Vorversion eines Festplatten-Backup-Programms sowie, neben weiteren, nützlichen Routinen, ein kleines Terminalprogramm enthalten sind, lohnt sich diese Diskette durchaus zum Kauf.

Eine Sammlung kleinerer Demo- und Hilfsprogramme bekommt der Käufer der Fish »98«. Beispielsweise eine Demoversion des Shareware-Disksorters »DiskCat«, das über umfangreiche Möglichkeiten verfügt. Leider kann die Demoversion nur maximal 100 Einträge speichern. Zum Reinschmecken und neugierig werden ist das Demo aber allemal gut. Des weiteren findet sich ein Backup-Utility, das DFÜ-Programm »Access!«, eine Mini-Dateiverwaltung sowie verschiedene Hard-Disk-Routinen und ein Quizprogramm. Alles in allem eine Diskette, die zur Vervollständigung der PD-Sammlung in Betracht gezogen werden sollte.

Der Grafik-Pool

Grafik-Spezialisten werden von der Fish »99« begeistert sein. Diese PD-Diskette enthält das Programm »A-Render«, mit dem sich Raytracer-Bilder berechnen lassen. Außerdem finden sich einige bereits berechnete Demo-Bilder mit auf dieser lohnenswerten Diskette. Wir meinen: Diese Diskette ist für grafikbegeisterte Amiga-Fans ein Muß.

Nicht notwendig, aber doch sehr schön anzusehen ist ein mit Videoscape 3D erstelltes Animations-Demo, das auf Fish »100« enthalten ist. Es stellt in brillanter 3D-Qualität einen Amiga-Ball dar, der drei detailliert gezeichnete Einräder »balanciert«. Dieses Demo, das wundervoll die Leistung von Videoscape demonstriert, ist nur auf Amiga-Modellen lauffähig, die über mindestens 1 MByte Hauptspeicher verfügen. Für Grafik-Freaks sicher eine Abwechslung.

Wollten Sie sich schon immer mal auf der Diskette umsehen, Sektoren oder die ganze Diskette verändern? Dann ist Fish »102« genau das Richtige für Sie! Mit »Sectorama« (Bild) haben Sie vollen Zugriff auf alle Sektoren der Diskette. Und das beste: Mit diesem Programm können Sie auch auf Festplatten zugreifen, was mit den gängigen Diskettenmonitoren leider bis heute nicht der Fall war. Das Programm besticht zwar nicht durch luxuriöse Leistungsdaten und weist auch ein paar Fehler auf (der Bildschirmaufbau wird nicht korrekt erledigt), jedoch hat der Anwender nun endlich die Chance, auf die Plattenlaufwerke zuzugreifen. Wir meinen: Diese Diskette ist ein Muß für jeden, der sich etwas näher mit den Daten auf der Diskette befassen möchte und auch eine Möglichkeit benötigt, um auf Festplatten-Daten zugreifen zu können. (dm)

Anbieter von Public Domain-Disketten (ohne Anspruch auf Vollständigkeit):
Atlantis, Dunantstr. 53, 5030 Hürth, Tel. 02233/31066
Datentechnik Bittendorf, Postfach 100248, 6360 Friedberg, Tel. 06031/61950
Stefan Ossowski, Veronikastr. 33, 4300 Essen, Tel. 0201/788778
Fastworks, Fichtestr. 16, 5090 Leverkusen 1, Tel. 0214/92802
EcoSoft AG, Kaiserstr. 21, 7890 Waldshut, Tel. 07751/7920
PD-Shop, Opladener Str. 30, 4018 Langenfeld, Tel. 02173/13657
Christian Bellingrath, Trift 10, 5860 Iserlohn, Tel. 02371/24192
Uwe Schmielewski, Haroldstr. 71, 4100 Duisburg 1, Tel. 0203/376448
Soyka Datentechnik, Hattinger Str. 685, 4630 Bochum 5, Tel. 0234/411913
Rainer Wolf Soft- und Hardwareversand, Deipe Stegge 187, 4420 Coesfeld, Tel. 02541/2874
AB Computersysteme, A. Büdenbender, Wildenburgstr. 21, 5000 Köln 41, Tel. 0221/4301442
C.A.S., Sprendlinger Landstr. 71, 6050 Offenbach, Tel. 069/842013
Jumbo Soft, Horemansstr. 2, 8000 München 19, Tel. 089/1234065

AMIGA INTERN.



Amiga-Anwender, die ihren Superrechner von Grund auf verstehen wollen, haben alle mit den gleichen Problemen zu kämpfen: Sie müssen sich durch mehrbündige, ausländische Bände arbeiten, die zudem nicht sonderlich aktuell sind. Oder aber sie besitzen den Ehrgeiz, die Geheimnisse ihres Rechners auf eigene Faust zu ergründen – einzige Voraussetzung hierbei: Man hat ein paar Jährchen Zeit. Einfacher ist es da schon mit dem neuen Amiga Intern: Hier finden Wißbegierige alles über den Amiga. In einem Band, in deutscher Sprache und absolut „up to date". Wie sehr Sie hier bis ins kleinste Detail vordringen, vermittelt bereits ein kurzer Blick ins Inhaltsverzeichnis: die Hardware des Amiga mit ausführlicher Beschreibung des 68000-Prozessors, der CIAS, der Customchips und der Schnittstellen, die Programmierung der Hardware in Assembler mit Speicherbelegung, Interrupts, Grafik und Sound. Doch Amiga Intern würde diesen Namen nicht verdienen, wenn es zusätzlich nicht noch jede Menge Fakten und Details zum Betriebssystem liefern würde. So erfahren Sie alles Wissenswerte zur EXEC: die wichtigsten Strukturen, Funktion und Arbeitsweise des Multitasking, I/O-Handhabung und Verwaltung der Ressourcen, EXEC-Base, resetfeste Programme und Strukturen. Fehlt nur noch das AmigaDOS. Auch hierzu nur einige Stichworte: Funktionen, Parameterübergabe, Fehlermeldungen, Boot-Vorgang, Aufbau der Diskette, IFF-Format, Programmstart von CLI und Workbench, interner Aufbau der CLI-Befehle, die Devices. Selbstverständlich finden Sie auch zahlreiche Beispiele zur Programmierung der EXEC- und DOS-Routinen. In Amiga Intern finden Sie den kompletten Rechner zwischen zwei Buchdeckeln.

Amiga Intern
Hardcover
639 Seiten, DM 69,–

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort



ELECTRONIC ARTS™

ELECTRONIC ARTS DELUXE-Serie

Exklusivvertrieb bei Markt & Technik



NEU DELUXE
Paint II/deutsch für den Amiga

Deluxe Paint II ist eines der außergewöhnlichsten Grafikprogramme auf dem Softwaremarkt. Im Vergleich zu seinem Vorgänger ist Deluxe Paint II in der Lage, in der Textfunktion Fett- und Kursivschrift sowie Unterstreichungen einzubauen, perfekte Manipulation in der dritten Dimension darzustellen oder wichtige Parameter wie Längsdruck, Schwarzweißgraustufen - oder Farbdruck, Ränder und vieles mehr unmittelbar vor dem Druck per Mausklick zu verändern.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52584
DM 249,–* (sFr 225,–*/öS 2490,–*)

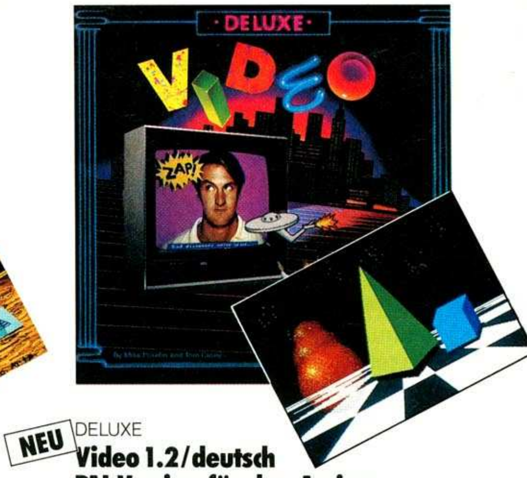
Jetzt lieferbar:
Music Construction Set
Das professionelle Musikprogramm für Ihren Amiga!
Hardware-Anforderung: Amiga 512 Kbyte.
Bestell-Nr. 52569
DM 249,–* (sFr 199,–*/öS 2490,–*)



NEU DELUXE
Print II/deutsch für den Amiga

und ein grafikfähiger Drucker verwandeln den Amiga in eine Druckmaschine. Sie können Karten, Poster, Briefköpfe und vieles mehr auf einfachste Weise entwerfen und ausdrucken. Besitzer eines Farbdruckers können ihr Werk auch in Farbe aufs Papier bringen. »Deluxe Print« ist kompatibel zu »Deluxe Paint«. Das bedeutet, daß man Grafiken zwischen den Programmen austauschen kann.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52582
DM 199,–* (sFr 179,–*/öS 1990,–*)

* Unverbindliche Preisempfehlung



NEU DELUXE
Video 1.2/deutsch PAL-Version für den Amiga

dient zum einfachen Entwerfen und Zusammenstellen von animierten Grafik-Sequenzen. Sie können so Videofilme mit Computergrafik versehen und regelrechte Computer-Videoclips zusammenstellen.
Das Programm ist ebenfalls kompatibel zu »DELUXE PAINT« und »DELUXE PRINT«.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52583
DM 249,–* (sFr 225,–*/öS 2490,–*)

Die ideale Ergänzung:
Das Buch zur Software
Markus Breuer
DELUXE Grafik mit dem Amiga
1986, 370 Seiten
ISBN 3-89090-412-2
Best.-Nr. 90412
DM 49,– (sFr 45,10/öS 382,20)
Erhältlich bei Ihrem Buchhändler



Deluxe Paint, Deluxe Print und Deluxe Video erhalten Sie in den Fachabteilungen der Kaufhäuser, in Computershops, bei Commodore-Fachhändlern oder direkt beim Verlag gegen Vorauskasse.

710228

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 67 75 26 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 48 15 43-0.

IFF: Der Grafikstandard

Der Amiga hat anderen Heim- und Personal Computern nicht nur seine Leistungsfähigkeit voraus. Er ist auch der erste Heimcomputer, auf dem es einen Standard zur Speicherung von Grafikbildern gibt. Hier erfahren Sie dessen Aufbau.

Nachdem eine große Resonanz der Leser auf den Grafikartikel im AMIGA-Heft 10/87 zu verspüren war, tauchen wir nun tiefer in die für viele noch unbekannte Welt des »Interchange File Format« (IFF) von Electronic Arts ein. Insbesondere wird hier der Teilbereich »InterLeaved BitMap« (ILBM) besprochen, da dies der am häufigsten verwendete Standard auf dem Amiga ist. Alle Grafiken auf dem Amiga werden in diesem Format gespeichert, so daß die Kompatibilität all dieser Programme untereinander kein großes Wunder mehr ist. Andere IFF-Datentypen seien hier nur kurz angesprochen: Es gibt noch »8SVX« (8 Bit Sampled Voice) für digitalisierte Musik, »FTXT« (Formatted TeXT) für Texte, »SMUS« (Simple MUsical Score) für nicht-digitalisierte Musikstücke und seit neuestem »ANIM« für Animations-Dateien. Der Aufbau einer IFF-Datei ist dank des Modul-Konzeptes bei allen Typen fast gleich. Den Typ einer Datei erkennt man in den ersten Bytes der Daten. Dort stehen dann obige 4stelligen Bezeichnungen, doch dazu gleich mehr.

Zurück zu ILBM, inzwischen einer der weitverbreitetsten Grafikstandards überhaupt. Es gibt wohl auf dem Amiga kaum Programme (schon gar keine käuflichen) die mit Grafiken und Bildern arbeiten und diesen Standard nicht unterstützen. Für alle, die sich mit der Programmierung der IFF-Strukturen genauer auseinandersetzen wollen, gibt es von Electronic Arts eine Public Domain-Diskette mit Erläuterungen und C-Sources. Diese Diskette ist als Amiga-Library 64, auch bekannt als Fish-Disk 64, im Public Domain-Versand erhältlich. Damit fällt es leichter, diesen Standard, der in Anlehnung an ein ähnliches XEROX-System entwickelt wurde, für eigene Lade- und Speicherroutinen zu verwenden, die sofort in eigene Programme eingebunden werden können. Zu dieser Diskette gibt es auch eine fast 100seitige Dokumentation, die sich ebenfalls jeder Interessierte kopieren darf. Darin werden alle Aspekte des IFF-Standards ausführlich und mit Beispielen erläutert. Einziger Haken dabei ist die leider manchmal schwer verständliche englische Sprache.



Bild 1. Der Fantasy Rider aus unserem Malwettbewerb ist das Demo-Bild, auf das sich dieser Artikel bezieht

Die zweite Möglichkeit ist, sich diesen Artikel durchzulesen, an dessen Ende Sie zwar kein fertiges Programm, aber doch die Kenntnis haben werden, ein solches zu schreiben.

Als Beispiel verwenden wir den »Fantasy Rider« von Jens Eisert aus unserem Malwettbewerb (Bild 1). Um die einzelnen Daten zu untersuchen, haben wir am Anfang die ersten 256 Byte dieses Bildes als Auszug aus einem Diskettenmonitor dargestellt (Bild 2). In der linken Spalte stehen, abgetrennt durch einen Doppelpunkt, die Adresse des ersten Bytes in einer jeden Reihe in hexadezimaler Form (von 0000: bis 00F0:). Diese Zahlen dienen nur der Orientierung und werden vom Diskettenmonitor mitausgegeben. Im großen vierspaltigen Block stehen die eigentlichen Daten, die reihenweise durchlaufen. Ganz rechts werden diese Daten nochmals, soweit möglich, in eine ASCII-Darstellung umgewandelt; dies ist ein Service des Monitors. Zu jedem Abschnitt, der besprochen wird, gibt es ein Bild (3a bis 3e). Darin werden Sie immer Teilbereiche dieser ersten 256 Byte sehen. Es sind jeweils die Bytes, um die es in diesem Abschnitt geht, unterlegt. Dunkler unterlegte Bytes (blau) enthalten den Chunk-Namen und die Längenbytes, hellere (gelb) die eigentlichen Daten. Alle IFF-Dateien sind aus sogenannten »Chunks« aufgebaut. Das sind kleine Abschnitte, die jeweils bestimmte Informationen beinhalten. Jeder Chunk hat einen Namen, der an dessen Anfang steht.

Der erste Chunk heißt »FORM« und ist, wie der Name schon sagt, für das Format der gesamten Datei zuständig (Bild 3a). Er enthält in seinen ersten vier Byte, wie jeder Chunk, seinen Namen im ASCII-Code. Diese Namen sind, wie die Dateitypen auch, immer genau vier Byte beziehungsweise Zeichen lang. Danach folgt in einem Langwort (besteht aus vier Byte) die ge-

Der Form-Chunk

samte Länge dieses Chunks. Beim FORM-Chunk steht hier nicht nur dessen Länge, sondern die Länge aller Chunks dieser Datei minus 8. Die 8 wird abgezogen, da der Name und die Längenbytes des Form-Chunks nicht mitgezählt werden. In unserem Beispiel steht in diesen Bytes $EB96, was dezimal 60310 entspricht. Addiert man nun 8 hinzu, erhält man 60318, was genau der angezeigten Länge des Files im LIST-Befehl des CLI entspricht.

Sollte es einmal vorkommen, daß ein Chunk-Inhalt eine ungerade Anzahl von Bytes beinhaltet, muß zum Inhalt noch ein Null-Byte hinzugefügt werden, da Chunks immer nur an geraden Adressen beginnen dürfen. Dieses Null-Byte darf aber nicht zur Chunk-Länge addiert werden, da diese immer nur die wirkliche Anzahl an Datenbytes angibt. Solche Sonderfälle kommen aber nicht in den Standard-IFF-Chunks vor.

```
0000: 464F524D 0000EB96 494C424D 424D4844    FORM....ILBMBMHD
0010: 00000014 01400190 00000000 05020100    .....@..........
0020: 0000140B 01400190 434D4150 00000060    .....@..CMAP...`
0030: 00000020 10003020 00505060 D0D0E0C0    ... ..0 .PP`....
0040: C0D0B0B0 C0A0A0B0 9090A070 70804030    ..........pp.@0
0050: 10504020 60503070 60408070 50908060    .P@ `P0p`@.pP..`
0060: A09070B0 A080D0C0 A0F0F0F0 40303050    ..p.........@00P
0070: 40406040 40704040 80504090 6040A070    @@`@@p@@.P@.`@.p
0080: 50B08050 D09050E0 A050F0C0 50F0E0A0    P..P..P..P..P...
0090: 43524E47 00000008 00000000 00000000    CRNG............
00A0: 43524E47 00000008 00000000 00000000    CRNG............
00B0: 43524E47 00000008 00000000 00000000    CRNG............
00C0: 43524E47 00000008 00000000 00000000    CRNG............
00D0: 424F4459 0000EAC5 091FFDE3 FBC00000    BODY............
00E0: A20704FE 001A0300 204F879F 8002EF81    ........ O......
00F0: 8BFCFFE2 7FC5BF8F C3FDF001 E009B000    ................
```

Bild 2. Ein Speicherauszug der ersten 256 Byte des Bildes »Fantasy Rider«

```
0000: 464F524D 0000EB96 494C424D 424D4844     FORM....ILBMBMHD
```

Bild 3a. Die ersten 12 Byte der Grafik

Nach dem Chunk-Namen und dessen Länge folgt grundsätzlich der Chunk-Inhalt. Der besteht beim FORM-Chunk nur aus vier Byte, die den IFF-Typ (hier ILBM) angeben. Auch diese Buchstaben liegen natürlich im ASCII-Format vor.

Als zweiter Block folgt der »Bitmap-Header«-Chunk. (Bild 3b) Auch er beginnt mit seinem Namen (BMHD) und vier Byte, die seine Länge angeben (und diesmal wirklich nur die genaue Länge, allerdings ohne Namen und Längenbytes). Hier steht $14, was dezimal 20 entspricht. Der Inhalt des BMHD-Chunk ist also genau zwanzig Byte lang.

Die ersten Datenbytes des Chunk-Inhalts, die nun folgen, geben die Breite und die Höhe des Bildes an. Dafür werden jeweils Byteworte verwendet (zwei Byte lang). In unserem Demobild stehen hier die Werte $0140 und $0190, was einer Auflösung von 320 x 400 Punkten entspricht und auf den Interlace-Modus hindeutet.

Die nächsten zwei Byteworte geben die Position der Grafik auf dem Bildschirm an und sind (warum, wissen wohl nur die Erfinder des Standards) vorzeichenbehaftet. Zuerst wird hier die Pixelkoordinate in der vertikalen, dann in der horizontalen angegeben.

Bei Programmen wie Graphicraft oder Deluxe-Paint II sind diese Bytes immer auf Null gesetzt, da deren Grafiken immer an der Position 0/0 (also in der oberen linken Ecke des Bildschirms) beginnen.

Ein vorzeichenbehaftetes Byte rechnet man einfach um, indem man von 256 den dezimalen Wert des Bytes abzieht, als ob es nicht vorzeichenbehaftet wäre.

Als nächstes folgt das »nPlanes«-Byte, das die Anzahl der verwendeten Bitplanes angibt. Diese können bei Deluxe-Paint frei gewählt werden. Eine Bitplane ermöglicht die Verwendung von zwei Farben, zwei Bitplanes bieten vier Farben, drei Bitplanes acht Farben und so weiter. Insgesamt bieten n Planes zwei hoch n Farben. In unserem Beispiel steht bei nPlanes eine 5, woraus die Verwendung von 32 Farben ersichtlich wird.

Das folgende Byte gibt die Art der verwendeten Maskentechnik an. Diese Funktion ist etwas komplizierter. Eine Maske können Sie sich als Bitplane vorstellen, welche die gleiche Auflösung wie das eigentliche Bild hat und über diesem liegt. Eine Eins in dieser Masken-Bitplane gibt an, daß die Grafik dort sichtbar ist — eine Null, daß sie transparent ist. Transparent heißt in diesem Zusammenhang natürlich nicht durchsichtig. Vielmehr wird ein so markiertes Pixel nicht mehr als Pixel in einer der Vordergrundfarben angesehen, sondern gehört, welche Farbe es auch immer hat, zum Hintergrund. Diese Art der Vorder- oder Hintergrundmarkierung findet zum Beispiel bei der Stencil-Funktion von DPaint II Verwendung. Eine Anwendung wäre beispielsweise die Erstellung eines mehrfarbigen Hintergrundes, der von Zeichenprogrammen dann ausschließlich als Hintergrund erkannt wird. Steht in diesem »Masking«-Byte nun eine Null, wird in diesem Bild keine Maske verwendet. Eine Eins gibt an, daß die Maske als normale Bitplane dem »BODY«-Chunk (wird später erläutert) beigefügt wurde. Kommt in besagtem Byte eine Zwei zum Vorschein, wird automatisch mit einer Maske gearbeitet, in der alle Bits auf 1 gesetzt sind, die in der »Transparent Color« gezeichnet wurden. Diese »Transparent Color« wird später im Chunk noch definiert.

Der Bitmap-Header

Zu guter Letzt kann im Masking-Byte auch eine drei stehen, was eine sogenannte »Lasso-ing«-Funktion der Maske zur Folge hat. Dabei entsteht um das eigentliche Grafikbild ein Ein-Pixel breiter Rand, der dann nach innen ausgefüllt wird, bis das Programm auf eine andere Farbe stößt. Alle so erkannten Pixel sind dann »transparent« und zeigen den Hintergrund. Amiga-Programme, die diese Funktion voll ausnutzen gibt es noch nicht, aber auf dem Macintosh-Standard-Programm »MacPaint« ist diese Funktion ein fester Bestandteil der Bildspeicherung. Bei unserem Bild liegt hier eine standardmäßige 2 vor.

Weiter geht es mit dem »Compression«-Byte. Es zeigt an, ob die Grafik im komprimierten oder im Original-Zustand auf Diskette gespeichert wurde. Das Packen von Grafikdateien hat den Vorteil der schnelleren Zugriffszeit und spart Speicherplatz auf der Diskette. Dabei werden mehr als zwei gleichfarbige, nebeneinanderliegende Pixel zu nur zwei Byte zusammengefaßt. Den genauen Kompressionsalgorithmus erläutern wir später noch ausführlicher; an dieser Stelle sei nur erwähnt, daß eine Null auf den originalen und eine Eins auf den komprimierten Zustand hindeutet. Unser Demobild vom Reiter ist also gepackt (Compression-Byte steht auf eins). Als nächstes kommt ein Byte, das bis jetzt noch keinerlei Verwendung hat und deshalb für spätere Erweiterungen reserviert wurde. Um Konflikte mit weiteren Programmen zu vermeiden, wird dieses Byte immer auf Null gesetzt.

Es folgt das oben schon erwähnte Transparent-Color-Bytewort, welches die Nummer des Farbregisters enthält, das durchsichtig sein soll. Dieses Byte ist natürlich nur dann relevant, wenn im Masking-Byte eine Zwei oder eine Drei eingetragen ist. Ansonsten wird es auf Null gesetzt. Eine Null kann aber auch, wie bei unserem Beispiel, die Transparenz der Farbe im Register Null bedeuten, die bei Deluxe-Paint normalerweise schwarz ist.

Nun folgen zwei Byte, die das Verhältnis der Pixelbreite zur Pixelhöhe angeben. Dieses ist bei der verwendeten Auflösung von 320 x 400 Punkten 20:11. Bei der niedrigen Auflösung von 320 x 200 wäre dies 10:11.

Die zwei folgenden und letzten Byte-Worte des BMHD-Chunks geben die eigentliche Bildschirm-Größe an (vorzeichenbehaftet!). Das ist wichtig, da Grafiken auf dem Amiga auch größer sein können als der definierte Bildschirm. Dies ist beim Laden einer Overscan-Grafik in den Normalmodus von Bedeutung. Üblicherweise stehen aber hier dieselben zwei Byteworte wie schon am Anfang des BMHD-Chunks, der die Auflösung der Grafik angab; so auch bei unserem Bild vom Fantasy Rider.

```
0000: 464F524D 0000EB96 494C424D 424D4844     FORM....ILBMBMHD
0010: 00000014 01400190 00000000 05020100     .....@.........
0020: 0000140B 01400190 434D4150 00000060     .....@..CMAP...`
```

Bild 3b. Die Byte des Bitmap-Headers

Die Farbpalette

Nun folgt ein neuer Chunk mit der Bezeichnung »CMAP« (Bild 3c). Er ist für die Speicherung der Farbpalette (Colormap) zuständig und enthält die Informationen von bis zu 32 Farben.

Nach dem Chunk-Namen und der obligatorischen Länge folgen nun für jede Farbe drei Byte, die die Rot-, Grün- und Blauanteile der Farbe angeben. Diese Werte sind aber jeweils im höherwertigen Teil des Bytes, also in den Bits vier bis sieben, abgelegt. Für die Farbe Schwarz würden alle drei Bytes auf Null stehen. Für ein kräftiges Rot müßte das erste Byte auf $F0 und die restlichen zwei auf Null stehen und so weiter. Sie merken, daß sich zur Darstellung dieses Formats das hexadezimale Zahlensystem hervorragend eignet.

Durch die Verwendung des oberen Nibbles (so heißt ein »halbes« Byte) ergeben sich pro Farbanteil 16 Kombinationsmöglichkeiten ($00, $10, $20, $30 und so weiter). Es sind also insgesamt 16 x 16 x 16 = 4096 Farben erreichbar. Dies ist genau das, was uns ja auch schon die Hardwarebücher zum Amiga lehrten.

Sicherheitshalber sollten Sie immer, auch wenn Sie nur vier Farben verwenden, alle 32 Farben speichern und die nichtbenutzten Farbregister auf Null setzen.

Nach den 32 x 3 = 96 Byte, die auch als Chunk-Länge angegeben sind, ist auch dieser Chunk beendet.

Der Farbwechselbereich

Als letzter Block (Bild 3d) vor dem eigentlichen Grafikbild folgt der CRNG-Chunk (Color Register Range). Er enthält Daten, die Programme wie Deluxe-Paint zu kleinen Farbanimationen (auch als Color-Cycling bekannt) verhelfen. Dieser CRNG-Chunk wird pro Grafikbild viermal wiederholt, wobei das erste Vorkommen für die Shade-Funktion von Deluxe-Paint vorbehalten und die restlichen drei Chunks für normales Farbrollen gedacht sind.

Wir machen Preise!

Sie zahlen: keine Einfuhrumsatzsteuer, keine Verpackung, keine Versandspesen

HITS DES MONATS

DIGI VIEW	289,–
DIGI PAINT	98,–
VIDEOSCAPE 3D	279,–
ZING!	149,–
ZING KEYS	89,–
SILVER	279,–
TEST DRIVE	69,–
FIRE POWER	59,–
GARRISON	59,–
TERRORPODS	59,–

ANIMATION

	DM
AEGIS Animator & Images	269
APPRENTICE 3d-Animator	498
APPRENTICE JUNIOR 3D-Animator	129
APPRENTICE Libraries: GEO	79
APPRENTICE Libraries: LETTERS	79
APPRENTICE Libraries: NUMBERS	69
DELUXE VIDEO II	198
PAGE FLIPPER deutsch	89
SILVER Ray Tracing Animator	279
VIDEOSCAPE 3D	279

BÜCHER

	DM
ADVANCED AMIGA BASIC	49
ADVANCED C PRIMER	69
AMIGA 1st COMPUTER	49
AMIGA APPLICATIONS	49
AMIGA BASIC IN&OUT BOOK	59
AMIGA C FOR BEGINNERS	49
AMIGA Jahrbuch 1987	10
AMIGA PROGRAMMERS GUIDE	49
AMIGA TRICKS&TIPS BOOK	49
AMIGA USER'S GUIDE	49
BEGINNERS GUIDE AMIGA	49
C PRIMER PLUS	69
C PROG. GRAPHICS BOOK	59
DOS Manual Book	59
ELEMENTARY AMIGA BASIC	49
FLIGHT SIM. ADV. BOOK	49
INSIDE AMIGA GRAPHICS	49
INSIDE THE AMIGA	69
KIDS AND THE AMIGA	49
MASTER AM DOS Buch	39
THE AMIGA BOOK	59
USING AMIGA DOS	49

DATENBANK

	DM
ACQUISITION	598
dBMan dBase III komp.	289
MICRO FICHE FILTER	189
ORGANIZE database	189
SUPERBASE Personal	198

DEMOS

	DM
DEMO Acquisition	12
DEMO DIGI-VIEW	12
DEMO DisCovery	12
DEMO Dynamic CAD	12
DEMO EASYL Bilder	12
DEMO Gridiron	12
DEMO King of Chicago	12
DEMO Perfect Sound	12
DEMO POWER WINDOW & TxED	12
DEMO PRO WRITE & FLOW	12
DEMO VIZAWRITE Desktop	12
FUNKTION	12

DIVERSES

	DM
AEGIS DIGA!	149
BUSSINESS CARD MAKER	119
FLIPSIDE printer driver	98
FRED FISH Disks je	5
FUNKTION Graphenzeichner dt.	79
KICK START ELIMINATOR	198
PUBLIC DOMAIN Disks je	5

FESTPLATTEN

	DM
20 Meg. Hard Drive SCSI	1555
50 Meg. Hard Drive SCSI	2595
Harddisk 20 MB für A-2000/PC	998
Harddisk 20 MByte für A-2000	1498

GRAFIK

	DM
AEGIS DRAW	149
AEGIS DRAW PLUS	449
AEGIS IMAGES	79
AEGIS IMPACT	149
CALLIGRAPHER	209
DELUXE PAINT II	229
DELUXE PRINT & Data #1	139
DIGI PAINT	98
DYNAMIC CAD	898
FORMS IN FLIGHT 3D CAD Stereo	149
PRINTMASTER PLUS	98
PRISM	98
SCULPT 3D Ray-Tracing	179
TV-TEXT 3D Textanim.	198

GRAFIKDATEN

	DM
AEGIS Artpack-1	59
ART GALLERY #1	59
ART GALLERY #2	59
DONALD DUCKS PLAYGROUND	69
GRAFIKBILDERDISK #1	12

KABEL

	DM
AMIGA 1000 Druckerkabel	29
AMIGA Bildschirmkabel Scart	29
CENTR.Adap. 500/2000 Bausatz	29
CENTRONICS Adapter A500/2000	49

KALKULATION

	DM
ANALYTIC ART	98
ANALYZE 2.0 spreadsheet	298
LOGISTIX	298
MAXIPLAN PLUS Lotus komp.	398

LAUFWERKE

	DM
Laufwerk 3½ Zoll extern	348
Laufwerk 3½ Zoll intern	328

MUSIK

	DM
AEGIS AUDIOMASTER	98
AEGIS SONIX Musikprogramm	149
DELUXE MUSIC Construction Set	198
DNA MUSIC	69
DYNAMIC DRUMS	139
ENCORE mouse/keyb/macro	89
EYE PLAY 2	69
FRACTAL MUSIC	69
FUTURE SOUND II	337
INSTANT MUSIC	98
MUSIC STUDENT	98
PERFECT SOUND mit Digitizer	169
PRO MIDI STUDIO	298
PROTEIN MUSIC	69
SOUND SCAPE Sampler	179
STUDIO MAGIC	98
THE MUSIC STUDIO	98
WAVEBUILDER Supersound	69
WAVEFORM EASEL	69

MUSIKDATEN

	DM
60's GREATEST (45 min.)	49
70's GREATEST (45 min.)	49
80's GREATEST (50 min.)	49
BARBERSHOP (65 min.)	49
BEATLES Part 1 (40 min.)	49
BEATLES Part 2 (40 min.)	49
BEETHOVEN (40 min.)	49
BILLY JOEL (65 min.)	49
BROADWAY THEMES (40 min.)	49
CHRISTMAS (50 min.)	49
CHURCH MUSIC (50 min.)	49
CLASSICAL #1 (40 min.)	49
CLASSICAL #2 (40 min.)	49
CLASSICAL #3 (35 min.)	49
CLASSICAL #4 (30 min.)	49
CLASSICAL #5 (50 min.)	49
COMPOSER DISK Volume I	69
COMPOSER DISK Volume II	69
COMPOSER DISK Volume III	69
COUNTRY #1 (45 min.)	49
COUNTRY #2 (50 min.)	49
GOLD&PLATINUM (60 min.)	49
INSTR. DISK B-3 Organ	69
INSTR. DISK Melotron	69
INSTR. DISK Steeldrums	69
IT'S ONLY ROCK'N'ROLL	49
KENNY ROGERS (45 min.)	49
MOVIE THEMES (40 min.)	49
NOSTALGIA (45 min.)	49
POLKA PARTY (40 min.)	49
R. RODGERS SONGBOOK	49
ROCK Part 1 (50 min.)	49
ROCK Part 2 (50 min.)	49
SYMPHONY JUKEBOX	49
SYMPHONY MUSIC VIDEO	49
TV THEMES (35 min.)	49

SIMULATION

	DM
ACHELON Flugsimulator	129
FLIGHTSIMULATOR II	98
GALILEO Planetarium	149
JET Simulator	98
Scenery Disk #7	49
SUPER HUEY	59
SURGEON Operationssimulation	69

SPEICHER

	DM
COMSPEC 2 MByte RAM	998
Speicher 512KB & Uhr f. A-500	249
Speichererweiterung 2 MByte	795

SPIELE

	DM
A MIND FOREVER VOYAGING	89
ADVENTURE CONSTRUCTION	69
ALIEN FIRES adventure	69
ALT. REALITY 'THE CITY'	89
ARAZOK'S TOMB adv.	59
ARCHON game	89
ARCTICFOX game	69
ARENA & BRATACASS	79
AUTO DUEL sim.	119
BALANCE OF POWER	79
BALLYHOO mystery	79
BARBARIAN	69
BARDS TALE adv.	89
BEAT IT	29
BLACK CAULDRONS	98
BLACK JACK game	69
BORROWED TIME game	69
BOWLING	29
BRIDGE 4.0	89
BUREACRACY comedy	89
CHALLENGER	25
CHAMPIONSHIP BASEBALL	69
CHAMPIONSHIP BASKETBALL	69
CHAMPIONSHIP FOOTBALL	79
CHAMPIONSHIP GOLF	79
CHESSMASTER 2000 Schach	79
CHESSMATE Schach	49
COMPUTER BASEBALL	68
D'BUDDY	129
DECIMAL DUNGEON educ.	98
DEEP SPACE	79
DEFENDERS OF THE CROWN	79
DEJA VU	79
DELUXE MAPS	59
DEMOLITION	25
DETONATOR Spitzensound	69
DISCOVERY TRIVIA game	98
DR. FRUIT	29
DR. XES	98
EARL WEAVER BASEBALL	98
ENCHANTER adv.	59
FAERY TALE ADVENTURE	89
FIGHTER COMMAND	89
FINAL TRIP	29
FIRE POWER game	59
GALACTIC INVASION game	49
GALAXY FIGHT	29
GARRISON	59
GOLDEN OLDIES	59
GOLDEN PYRAMID	69
GRAND SLAM TENNIS	69
GRIDIRION	119
GUILD OF THIEVES Adv.	79
GUNSHIP game	86
HACKER II game	79
HALLEY PROJECT	69
HEX game	69
HEX graphic/strategy	89
HITCHHIKERS GUIDE Adv.	59
HOLLYWOOD HIJINX	69
Hollywood Poker	49
JEWELS OF DARKNESS	69
KAMPFGRUPPE strategy	109
KARATE	69
KARATE KID II	74
KARATE KING	49
KING OF CHICAGO game	89
KINGS QUEST 1	89
KINGS QUEST 2	89
KINGS QUEST 3	89
KNIGHT ORC adv.	79
LAND OF LEGENDS adv.	89
LEADER BOARD (Golf Game)	69
LEADER BOARD TOURNAMENT	39
LEATHER GODDESSES	79
LITTLE COMPUTER PEOPLE	59
LYBIANS IN SPACE	59
MAGICIANS DUNGEONS	79
MARBLE MADNESS	69
MIND BREAKER	29
MIND FOREVER	79
MINDSHADOW	69
MISSION ELEVATOR	79
MOONMIST	79
PACK BOY	29
PHALANX	25
PHANTASIE III Wrath of Nikad	98
POKER	79
PORTAL	79
RETURN TO ATLANTIS	109
ROADWAR 2000 game	98
ROGUE Adventure	68
SHADOWGATE	79
SHANGHAI	69
SHOOTING STAR	29
SILICON DREAMS	69
SINBAD game	79
SOLITAIRE	49
SPACE BATTLE	25
SPACE FIGHT	29
SPACE QUEST	98
STAR GLIDER game	79
STARFLEET I	89
STATIONFALL	79
STRIP POKER	69
TASS TIMES IN TONETOWN	79
TEMPLE OF ASPHAI	68
TENTH FRAME	79
TERRORPODS	59
TEST DRIVE	69
THE GRID	98
THE PAWN Adventure	79
TRINITY	79
TURBO car racing	49
ULTIMA III	89
UNINVITED	79
VADER	29
VIDEO VEGAS	59
WINTERGAMES	59
WISHBRINGER	39
WIZARDS CROWN	99
WORLD GAMES	59
ZORK Trilogy (I+II+III)	119

SPRACHEN

	DM
AC BASIC Compiler	398
AZTEC C Commercial	898
AZTEC C Developers	498
AZTEC C Personal	439
LATTICE C Compiler V 3.1	398
METACOMCO Lisp	398
METACOMCO Makro Assembler	169
METACOMCO Pascal (ISO)	189
MODULA-2 Commercial	598
MODULA-2 Developer	298
MODULA-2 Standard	198
PASCAL UCSD	198

SYSTEME

	DM
AMIGA 2000 Grundgerät	2395
AMIGA 500 Grundgerät	1149

TEXT

	DM
CITY DESK Desktop P.	298
DESKTOP ARTIST	79
DYNAMIC WORD	298
FAST FONTS	79
FLOW V1.02 Idea Proc.	149
ORDER Desktop organizer	89
PAGE-SETTER (Umlaute)	228
PAGESETTER LASERSCRIPT	89
PRO WRITE	198
PUBLISHER 1000	349
SCRIBBLE	189
TALKER wordprocessor	149
VIZAWRITE Desktop dtsch.	179
VIZAWRITE DESKTOP engl.	189
ZUMA FONTS Vol. 1	59
ZUMA FONTS Vol. 2	59
ZUMA FONTS Vol. 3	59

TOOLS

	DM
BUTCHER GRAPHIC-CONV. + EDITOR	69
CLI MATE Diskutility	69
DEMONSTRATOR	69
DisCovery Disk Editor deutsch	198
DISCOVERY-SHELL	119
DISK TO DISK	98
DOS EXPRESS	59
DOS to DOS	98
FLOPPY ACCELERATOR	59
GIZMOZ 2.0	128
GRABBIT	59
KEY-GENIE	119
MARAUDER II	69
METACOMCO Shell	98
METACOMCO Toolkit	89
MIRROR HACKER PACKAGE	98
MIRROR Kopierprogramm	98
POWER WINDOWS C-Libs & Help	149
QUICK NIBBLE Duplicator	79
THE EXPLORER Debugger	98
TxED Editor	79
WOW disk cache system	79
ZING KEYS	89
ZING! V1.02	149

VIDEOZUBEHÖR

	DM
DIGI-VIEW Digitizer	289
aMega Board 1 MB Fast RAM	695
BASIC KEYBOARD OVERLAY	19
DOS KEYBOARD OVERLAY	19
EASYL 500 Zeichentablett	698
EASYL 1000 Zeichentablett	798
EASYL 2000 Zeichentablett	898
Erweiterungskit AT	1695
Erweiterungskit XT	1098
HOST ADAPTER SCSI	495
JITTER-RID Filterglas	29
Joystick CONTROLLER	14,95
Joystick J-1000 SUPERMICRO	19
Joystick STARFIGHTER HM	24
Joystick STARMASTER	19,80
Joystick TAC 2 Heavy Metal	29
MD-120 3½" Teakdiskbox	79
MD-60 3½" Teakdiskbox	59
MD-60L 3½" Teakdiskbox	79
MOUSE HIDE Leder Pad	29
MOUSE PAD (9½"★11")	18
PAL Video Karte AMIGA 2000	198
PRO MIDI INTERFACE	98
REFERENCE PAK (5 Cards)	39
TIME SAVER dtsch.	178
TRACKBALL	89
TV Modulator für AMIGA	59

Preisänderungen und Irrtümer vorbehalten

amigaland
a. koppisch
hohenwaldstraße 26
6374 steinbach

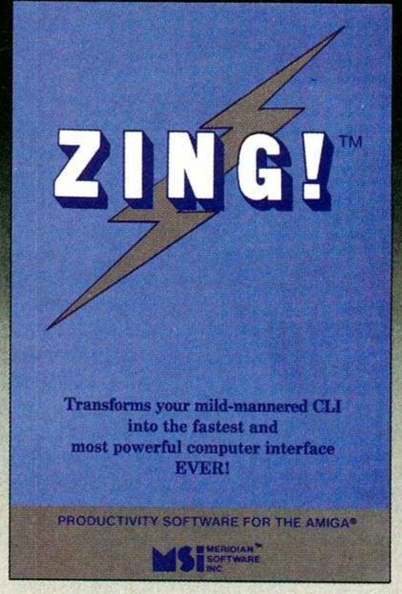
Bestellservice:
06171/71846

Alle Preise sind Endpreise.
Mindestbestellwert DM 50,-.
Versand ausschließlich gegen Nachnahme oder Vorkasse.

NEU!

AMIGA-Software

unentbehrlich für jeden Amiga-User – stop –
frisch bei Markt & Technik eingetroffen – stop –
deutsche Programmversionen in Arbeit – stop –
exklusiv bei Markt & Technik – stop –
Update-Service für alle unsere Kunden – stop



Das mächtige CLI-Werkzeug

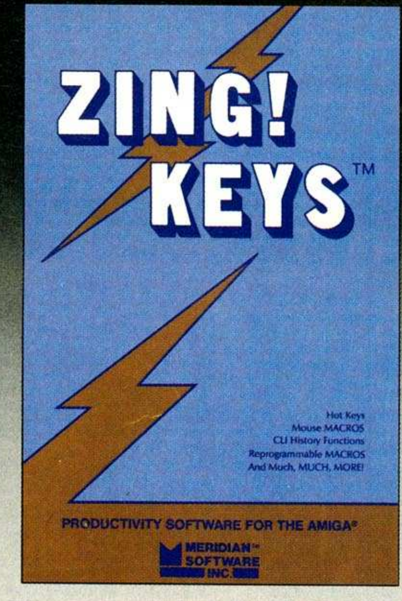
Mit ZING! haben Sie endlich das gesamte File-System mit Directories und Subdirectories fest im Griff. ZING! beschleunigt alle nötigen Arbeiten mit Files, verwaltet bis zu 500 Files und Subfiles und bis zu 100 Directories auf einmal. Die Bedieneroberfläche ist vom Feinsten:
- Pull-down-Menüs,
- (Click-)Icons,
- Funktionstasten.

Weitere Optionen wie: Task-Monitor, Printer-Spooler, Screen-Saver/Printer, Screen-Dimmer, Veränderung der Voreinstellung der Funktionstasten und des Systems. Am besten gleich bestellen!
Best.-Nr. 52571
DM 189,–* sFr 169,–/öS 1890,–*

Weitere Amiga-Software:

Superbase
Das relationale Datenbanksystem.
Best.-Nr. 51636
DM 249,–* sFr 199,–/öS 2290,–*

DELUXE Paint II/deutsch
Nutzen Sie die fantastischen Grafikmöglichkeiten Ihres Amiga – mit Deluxe Paint II.
Best.-Nr. 52584
DM 249,–* sFr 225,–/öS 2490,–*

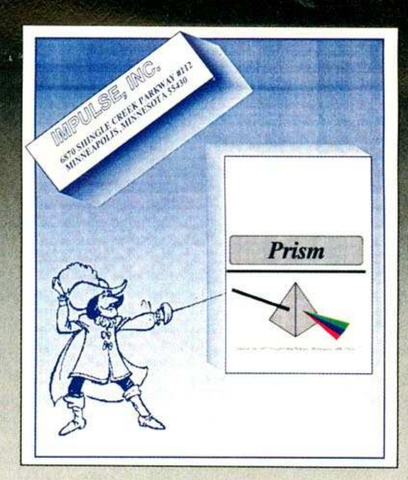


Ihr ganz persönlicher Amiga

Mit ZING! KEYS machen Sie aus Ihrem Amiga das variable System, das Sie sich schon immer wünschten. Es ist Ihren eigenen Ansprüchen jederzeit anpaßbar! Alle Tasten sind nach Wunsch belegbar: z.B. mit Funktionsaufrufen, Programmaufrufen, Systembefehlen und vorprogrammierten Befehlen. Die Belegung ist natürlich jederzeit abspeicherbar.
Durch die Belegung von »Hot-Keys« haben Sie mit ZING! KEYS ein Multitaskingsystem par excellence!
Best.-Nr. 52572
DM 109,–* sFr 98,–/öS 1090,–*

Weitere Amiga-Software:

DELUXE Print II/deutsch
Mit diesem Programm und einem grafikfähigen Drucker verwandeln Sie Ihren Amiga in eine Druckmaschine.
Best.-Nr. 52582
DM 199,–* sFr 179,–/öS 1990,–*



Das einzigartige 4096-Farben-Grafikprogramm

Prism ist nicht nur ein Zeichenprogramm – Prism ist mehr. Denn Prism beschränkt sich nicht auf die übliche 32-Farben-Palette.
Mit Prism haben Sie die ganze Bandbreite der 4096 Farbschattierungen des Amiga zur Verfügung. Auf einmal und in einem Bild! Wirklich einzigartig für digitalisierte Bilder und deren Manipulation. Und Sie können bis zu 4064 neue Farbtöne zu Ihren bestehenden Grafiken hinzufügen, Ausschnitte von einem Bild auf ein anderes übertragen und, und, und…
Best.-Nr. 52570
DM 159,–* sFr 142,–/öS 1590,–*

Weitere Amiga-Software:

DELUXE Video 1.2/ PAL-Version/deutsch
Entwerfen und Erstellen Sie animierte Grafiksequenzen, die Sie in Videofilme einbauen können! Alle drei DELUXE-Produkte sind untereinander kompatibel.
Best.-Nr. 52583
DM 249,–* (sFr 225,–/öS 2490,–*)

Musik Constructions Set
Das professionelle Musikprogramm für Ihren Amiga!
Best.-Nr. 52569
DM 249,–* sFr 199,–/öS 2490,–*

*Unverbindliche Preisempfehlung

Markt&Technik
Zeitschriften · Bücher
Software · Schulung

709230

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

```
0020: 0000140B 01400190 434D4150 00000060     .....@..CMAP...
0030: 00000020 10003020 00505060 D0D0E0C0     ... ..0 .PP`....
0040: C0D0B0B0 C0A0A0B0 9090A070 70804030     ...........pp.@0
0050: 10504020 60503070 60408070 50908060     .P@ `P0p`@.pP..`
0060: A09070B0 A080D0C0 A0F0F0F0 40303050     ..p.........@00P
0070: 40406040 40704040 80504090 6040A070     @@`@@p@@.P@.`@.p
0080: 50B08050 D09050E0 A050F0C0 50F0E0A0     P..P..P..P..P...
```

Bild 3c. Diese Farbpalette hat 32 Farben

```
0090: 43524E47 00000008 00000000 00000000     CRNG............
00A0: 43524E47 00000008 00000000 00000000     CRNG............
00B0: 43524E47 00000008 00000000 00000000     CRNG............
00C0: 43524E47 00000008 00000000 00000000     CRNG............
```

Bild 3d. Die vier Bereiche der Farbanimation

```
00D0: 424F4459 0000EAC5 091FFDE3 FBC00000     BODY............
00E0: A20704FE 001A0300 204F879F 8002EF81     ......... O......
00F0: 8BFCFFE2 7FC5BF8F C3FDF001 E009B000     ................
```

Bild 3e. Hier beginnen die eigentlichen Daten für die einzelnen Punkte der Grafik

Die einzige Aufgabe dieses Chunks besteht darin, einen Anfangs- und einen Endpunkt zu definieren, zwischen denen dann die Farben ständig ausgetauscht werden, was die Quasi-Animation zur Folge hat. Zusätzlich muß natürlich noch die Rollgeschwindigkeit angegeben werden.

Gibt man also als Eckpunkte die Farben 12 und 15 an, so wird bei der Animation Register 15 zwischengespeichert, der Inhalt von 14 nach 15, der von 13 nach 14 und der von 12 nach 13 kopiert. In Register 12 schließlich wird der zwischengespeicherte Wert kopiert und der Speicher gelöscht. Mit welcher Geschwindigkeit das alles geschieht, gibt das zweite Bytewort (1 Wort = 2 Byte) nach dem obligatorischen Namen und der Chunk-Länge an. Die ersten zwei Byte bilden wieder ein für spätere Anwendungen reserviertes Wort.

Ist im Geschwindigkeits-Byteworte der höchste erlaubte Wert (dezimal 16348 oder $4000) eingetragen, werden die Farben 60mal pro Sekunde ausgetauscht. Ist der Wert 8192 oder $2000, findet nur alle 30stel Sekunde ein Farbaustausch statt. Es besteht also ein proportionales Verhältnis zwischen dem Geschwindigkeitswert und der Austauschhäufigkeit pro Sekunde. Danach folgt wiederum ein Wort, das den Status der Animation bestimmt. Steht das Wort auf Null, findet keine Animation statt. Jeder andere Wert hat ein Farbrollen zur Folge.

Last but not least folgen zwei Byte, die die Nummer des unteren und die des oberen Registers enthalten, zwischen denen die Animation stattfinden soll. Steht hier das höhere Register zuerst, werden die Farben rückwärts gerollt.

Die »Cyclerate« und das Aktivierungswort sollten beim ersten CRNG-Chunk natürlich auf Null gesetzt werden, da die Shadefunktion von Deluxe-Paint hier kein Cycling unterstützt. Ansonsten haben alle Bytes die gleiche Wirkung. Bei unserer Demografik sind alle diese Werte auf Null gesetzt; es kommen also weder Farbanimationen noch Shadefunktionen vor. Damit wäre der Kopf eines Grafikbildes erläutert, kommen wir zum Hauptteil, dem schon erwähnten BODY-Chunk.

Ein Grafikbild besteht aus einer oder mehreren Bitplanes, die jeweils übereinander gelegt werden, damit für jedes Pixel ein oder mehrere Bits zur Verfügung stehen. Mit diesen Bits wird die Farbe bestimmt (Bild 3e). Bei vier übereinanderliegenden Bitplanes ergeben sich also pro Pixel vier Bit und damit 16 Kombinationsmöglichkeiten. Diesen Variationen sind durch die Farbregister eindeutige Farben zugeordnet (wenn man vom H.A.M.- und Halfbrite-Modus absieht), die das endgültige Erscheinungsbild des Punktes bestimmen. Bei anderen Computern werden nun die Bitplanes einfach hintereinander gespeichert, was aber den Nachteil hat, daß ein Bild immer komplett, also nicht ausschnittsweise, geladen werden kann. Man hat dieses Manko erkannt und im ILBM-Standard einer zeilenweisen Speicherung den Vorzug gegeben.

Dabei wird zuerst die erste Zeile der ersten Bitplane, dann die erste Zeile der zweiten Bitplane und die erste Zeile der dritten Bitplane gespeichert. Das geht bis zur ersten Zeile der n-ten Bitplane. Daran schließt sich, falls verwendet, die erste Zeile der Masken-Bitplane an. Dann folgt die zweite Zeile der ersten Bitplane, die zweite Zeile der zweiten Bitplane und so weiter, bis schließlich die letzte Zeile der n-ten Bitplane beziehungsweise die letzte Zeile der Maskenbitplane abgelegt wurde.

Das führt bei größeren Bildern unweigerlich zu einer Flut von Daten. Um diese Daten zu komprimieren (der Kompressionszustand wird in besagtem Byte im BMHD-Chunk festgelegt), verwendet man einen kleinen, aber dennoch effektiven Algorithmus. Die Kompaktierung darf logischerweise nur zeilenweise erfolgen.

Kommen wir zur Vorgehensweise des Packers: Das erste Byte wird als vorzeichenbehaftetes Befehlsbyte für den Packer interpretiert (n). Liegt n zwischen null und 127, werden die nächsten n+1 Bytes ohne Änderung übernommen. Das ist der Normalfall. Liegt n jedoch im Bereich von -1 bis -127, tritt der Packer in Aktion. Und zwar wird das nun folgende Byte -n+1mal wiederholt. Das hört sich zwar kompliziert an, ist aber schon alles, was der Packer macht. Daß dieser Vorgang einige Ersparnis bringt, kann man sich leicht ausrechnen. Eine Zeile schwarz mit einer Länge von 320 Punkten entspricht unter normalen Umständen 40 Byte. Der Packer erkennt dies und komprimiert diese Zeile der Bitplane auf zwei Byte ($D9 $00). Daß dies stimmt, läßt sich leicht nachrechnen. $D9 entspricht dezimal 217, was vorzeichenbehaftet -39 ist. -39 ist also in diesem Falle unser n, womit sich die Zeile nach der Formel -n+1 (-(-39)+1=40) wieder auf 40 Byte »aufbläht«.

Ein weiteres Beispiel: 14 Byte der Farbe Schwarz werden auf die Werte $F3 $00 komprimiert. $F3 entspricht 243, was vorzeichenbehaftet -13 ist. Nach der Formel -n+1 ergibt sich der Term -(-13)+1=14. Also eine 14malige Wiederholung des folgenden Bytes ($00).

Das dürfte zum Schreiben Ihrer eigenen Speicher und Lade-Routinen reichen. Wenn nicht, sei an dieser Stelle noch einmal auf die am Anfang erwähnte Dokumentation verwiesen, die Sie über Händler direkt bei Electronic Arts anfordern sollten.

Mit diesem Artikel haben Sie jedoch eine gute Grundlage zur Verarbeitung von ILBM-Dateien und können Ihre Programme auf den allgemeinen Standard zuschneiden.

(Ottmar Röhrig/jk)

Tabelle der Chunk-Belegungen

FORM-Chunk	BMHD-Chunk		CMAP-Chunk			CRNG-Chunks		BODY-Chunk	
Chunk Länge (l)	Chunk-Länge	(l)	Chunk-Länge		(l)	Chunk-Länge	(l)	Chunk-Länge	(l)
(= Gesamtlänge)	Höhe	(w)	Farbe0	Rot	(b)	Nullwort	(w)	Datenbytes	(b)
Chunk-Art (4b)	Breite	(w)	Farbe0	Grün	(b)	Geschwind.	(w)	.	
(= ILBM)	X-Position	(w)	Farbe0	Blau	(b)	Aktiviert	(w)	:	
.	Y-Position	(w)	Farbe1	Rot	(b)	untere Farbe	(b)	:	
	Anz. Bitplanes	(b)	Farbe1	Grün	(b)	obere Farbe	(b)		
	Masking	(b)	Farbe1	Blau	(b)				
	Compression	(b)	Farbe2	Rot	(b)				
	Nullbyte	(b)	.						
	Transp. Color	(w)	:						
	X-Aspect	(bl)	:						
	Y-Aspect	(b)							
	Page-Width	(w)							
	Page-Height	(w)							

(b) = Byte, (w) = Wort, (l) = Langwort

Bild 4. Die Chunk-Belegung eines ILBM-Files im Überblick

Der Preisbrecher

MPS 1500 C heißt der neue, preisgünstige Farbdrucker von Commodore. Das klingt vielversprechend. Wir haben getestet, was der Drucker am Amiga leistet.

Ein Farbdrucker fehlt den meisten Amiga-Besitzern zu ihrem Glück. »Wozu?«, der Einsatz der gesamten Palette von Mal- und Zeichenprogrammen lohnt sich erst, wenn Sie Ihre Bilder farbig zu Papier bringen können. Auch für den reinen Texter bringt der farbige Ausdruck Vorteile: Briefköpfe oder wichtige Textpassagen lassen sich mit unterschiedlichen Farben hervorheben.

Es gibt zwar schon einige leistungsstarke Farbdrucker, doch diese sind für viele Anwender unerschwinglich. Nicht so der MPS 1500 C (Bild 1). Mit einem besonderen Blick auf die Besitzer des Amiga 500 ist er von Commodore ins Programm genommen worden.

Gute Verbindung

Der Anschluß ist einfach. Sie benötigen ein Kabel vom Centronics-Port des Druckers zum Parallelport des Amiga. Da der Drucker für den Amiga konzipiert worden ist, bereitet dies keine Probleme. Zunächst müssen Sie sich jedoch entscheiden, in welcher Betriebsart Sie den Drucker betreiben wollen. Der MPS 1500 C beherrscht drei verschiedene Emulationen:
— Epson JX 80
— IBM Proprinter und
— IBM Grafik-Drucker

Die Auswahl erfolgt nicht über DIP-Schalter, sondern über einen Programmiermodus, in den Sie nach Einschalten des Druckers und gleichzeitigem Drücken der Tasten <Linefeed> und <Formfeed> gelangen. In diesem Modus werden auch die anderen Grundeinstellungen des Druckers nach dem Einschalten, beispielsweise der Zeilenabstand oder die Auflösung (EDV, NLQ), programmiert.

Im Anschluß an die Einstellprozedur werden die neuen Einschaltparameter auf Wunsch gespeichert (CMOS-Speicherung) und stehen nach jedem Reset des Druckers zur Verfügung. Die meisten Werte können beim Drucken über Escape-Sequenzen verändert werden.

Für den Amiga empfiehlt sich die Epson JX 80-Emulation, da ein entsprechender Treiber auf der Workbench zur Verfügung steht.

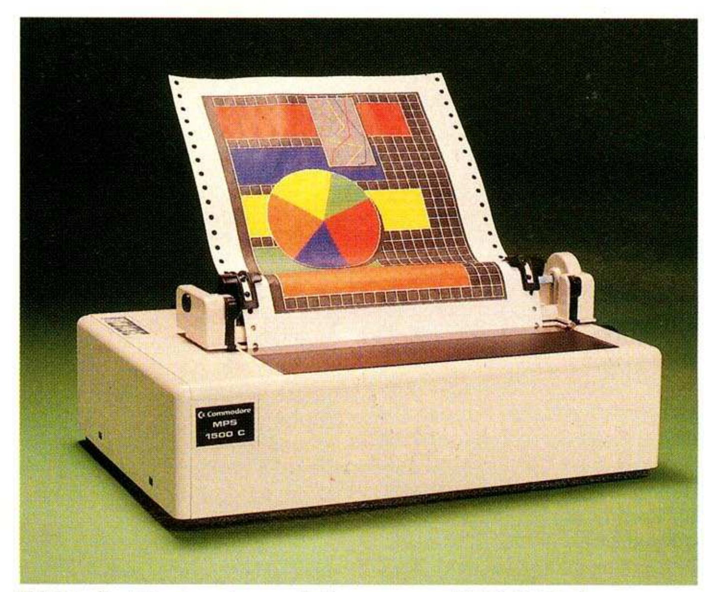


Bild 1. Farbig drucken mit dem neuen MPS 1500 C

Mit dieser Kombination können Sie sowohl Farbgrafiken als auch Texte auf dem Drucker ausgeben.

Die Qualität einer Farbhardcopy ist gut, wie der Ausdruck unseres farbigen Druckertestbildes (Bild 3) zeigt. Auch die Mischfarben werden durch mehrmaliges Überdrucken ausgezeichnet wiedergegeben. Allerdings ist die Qualität des Bildes stark abhängig von der Abnutzung des Farbbands. Nur etwa vier Hardcopies lassen sich mit einem neuen Farbband in einwandfreien Zustand ausdrucken. Danach beginnen die meisten der verwendeten Farben zu verschmutzen. Dies hängt auch damit zusammen, ob beim Ausdruck von Mischtönen zuerst mit den hellen Farben gedruckt wird. Es ist letztlich ein Problem des Druckertreibers. Für häufige Testdrucke, bei dem es nicht auf die Brillanz und Sauberkeit der Farben ankommt, empfiehlt sich auf jeden Fall ein zweites Farbband für den täglichen Gebrauch. Der jeweilige Austausch der Kassetten ist unkompliziert.

Für den reinen Textausdruck steht Ihnen zudem ein preiswerteres einfarbiges Band zur Verfügung. Im Textmodus — mit und ohne NLQ (Bild 2) — kann der MPS mit den gängigen 9-Nadel-Druckern mithalten. Die Druckgeschwindigkeit liegt mit gemessenen 79 Zeichen je Sekunde beziehungsweise 25 mit Schönschrift im üblichen Rahmen.

Leider zeigte sich bei unseren Tests ein kleiner Fehler in der Druckersoftware. Der MPS 1500 C akzeptiert, obwohl im Handbuch angegeben, in der Epson-Emulation die Umschaltung der Schriftauflösung nicht. Sie können vom Amiga aus nicht zwischen der NLQ- und EDV-Schrift umschalten. Da auch der Drucker hierzu keine direkte Taste zur Verfügung hat, müssen Sie den Drucker mit der schon erwähnten Methode mühsam umprogrammieren.

Einfacher geht es nur, wenn Sie Texte mit Hilfe der IBM-Emulation und dem MPS 1000-Treiber ausdrucken. Jetzt funktioniert die Umschaltung einwandfrei. Dennoch sollte Commodore diesen Fehler beheben.

Ein weiteres Manko ist der schwer zu handhabende Zugtraktor. Bei diesem Teil wurde an der falschen Stelle gespart. Er ist instabil, schlecht zu montieren und die Papierführungen schwer einzustellen. Ist er

Alle Daten auf einen Blick

Drucker: MPS 1500 C
Drucktechnik: Matrixdrucker 9 Nadeln
Zeichenmatrix: 9 x 9
NLQ-Matrix: 18 x 9
Hexdump: ja
Funktionstasten:
Online, Linefeed, Formfeed dienen auch zur Druckerprogrammierung
Schnittstelle: parallel (Centronics-like)
Gewicht: 4,2 Kilogramm
Abmessungen (B x H x T):
370 x 94 x 253 mm
Emulationen:
Epson JX 80,
IBM-Proprinter
IBM-Grafikdrucker

Geschwindigkeiten:
(Zeichen/s)

EDV (ohne <LF>):	
angegeben:	120
gemessen:	94
Probetext:	79
NLQ:	
angegeben:	25
gemessen:	25

Hardcopy: (Testbild (640 x 200):

s/w	244 s
Farbe:	405 s

Lautstärke: mittel
Puffer: ja 5,5 KByte
Papier: Einzel- und Endlospapier bis 254 mm Breite
Durchschläge: maximal 2 Stück

Zeichensätze: ASCII und IBM
Ladbarer Zeichensatz: ja (Puffer nur 2,5 KByte)
Schriftwahl:
Pica, Elite, Schmal, Proportional, Breit, Doppelt, Fett, Unterstrichen, Kursiv, Hoch und Tief
Grafikauflösungen:
(Punkte/Zeile) 480, 576, 640, 720, 960, 1920
Farbband: S/W:
Lebensdauer:
3,5 Mio. Zeichen
Preis: 29 Mark
Farbe: 39 Mark
Lebensdauer: 4 Hardcopies ohne Farbänderung



Bild 2. Gutes Schriftbild in Farbe und Schwarzweiß

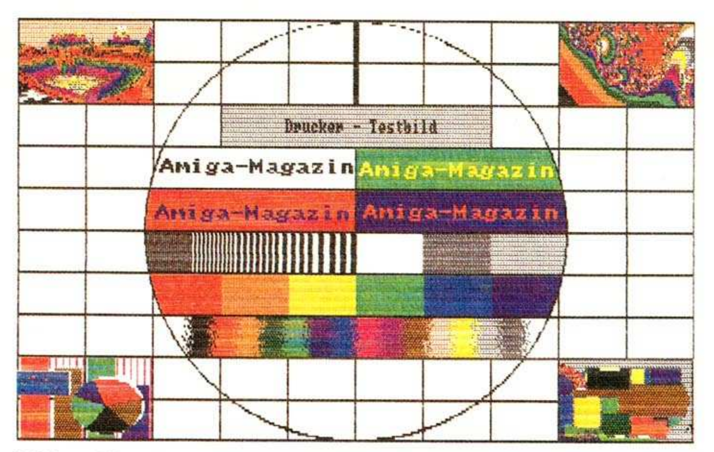


Bild 3. Eine Farbhardcopy — Das Bild wurde mit DeLuxe Paint II auf dem Amiga gemalt und mit dem Epson JX 80-Treiber gedruckt.

einmal korrekt eingestellt und muß nicht ständig abmontiert werden, können Sie vernünftig mit ihm arbeiten.

Außer Endlospapier verarbeitet der Drucker auch Einzelblattpapier. Dieses läßt sich ohne Probleme durch Drücken der Formfeed-Taste einziehen. Bei Verwendung von Einzelblättern verfügt der Drucker über die Fähigkeit des Papierrücktransports. Dies ist beim Zugtraktor unmöglich. Mit Hilfe des Rücktransports läßt sich der Druckkopf an jede Position des Blattes bewegen. Bei entsprechender Programmierung steht Ihnen somit ein DIN-A4-Plotter zur Verfügung. Die Genauigkeit, mit der Sie nach mehrmaligen Bewegungen des Druckkopfes an die Ausgangsposition gelangen, ist beim MPS 1500 C optimal.

Alles in allem ist es ein guter Farbdrucker mit kleinen Schwächen. Er bietet allen Besitzern des Amiga eine preiswerte Alternative zu den herkömmlichen Schwarzweiß- und den teuren Farbdruckern.

Der Preis

Apropos Preis. Der MPS 1500 C kostet 898 Mark. Damit liegt er deutlich unter der Schallmauer von 1000 Mark. Das bedeutet, ein Amiga 500 mit Farbdrucker ist zur Zeit für etwa 2000 Mark erhältlich.

AMIGA-WERTUNG

Hardware: Drucker Commodore MPS 1500 C

8,8 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■			
Verarbeitung	■	■	■			
Leistung	■	■	■	■	■	

Fazit: Der MPS 1500 C kann mit den gängigen 9-Nadel-Druckern konkurrieren, zumal wenn er in Farbe druckt.

Positiv: Drei Emulationen; guter Schriftabdruck und Grafikbild; sehr gute Plottereigenschaften bei Einzelblättern.

Negativ: Fehler in der Epson-Emulation; umständliche Programmierung; mangelhafter Zugtraktor

DATEN

Produkt: MPS 1500 C
Preis: 898 Mark
Hersteller: Olivetti
Anbieter: Commodore Fachhandel

Spätzünder sucht Marktlücke

AMIGA test

Wohl selten hat ein angekündigtes Programm so lange auf sich warten lassen wie Vizawrite für den Amiga. Jetzt endlich ist es fertig — wir zeigen Ihnen, ob sich das lange Ausharren für Sie gelohnt hat.

Mit dem abgewandelten Sprichwort »Was lange währt, wird endlich besser!« wirbt DTM, der Deutschland-Vertrieb von Vizawrite, für eine Textverarbeitung, die sich auf dem C 64 zum Klassiker entwickelte und deren Umsetzung für den Amiga bereits auf der CeBIT '86 in Hannover angekündigt wurde. Dennoch mußten sich Interessenten noch mehr als 12 Monate lang gedulden, bis das Programm jetzt, nach langen »Geburtswehen«, in den Handel kam.

Wird Vizawrite von der Workbench aus aktiviert, arbeitet es im 80-Zeichen-Modus des Amiga. Das entspricht einer Grafikauflösung von 640 x 256 Bildschirmpunkten (der PAL-Bereich wird voll unterstützt) und erlaubt ein relativ augenschonendes Arbeiten auch auf einem Farbmonitor. Wer jedoch einen lange nachleuchtenden Monochrom-Monitor sein eigen nennt oder seinem Augenarzt etwas Gutes tun möchte, der aktiviert Vizawrite mit einem erweiterten Befehl vom CLI aus und landet im Interlace-Modus. Dessen hohe Auflösung von 640 x 512 Pixel wird bekanntlich durch eine Halbierung der Bildwiederholfrequenz erreicht, so daß das Auge bei einer wenig nachleuchtenden Bildröhre ein unangenehmes und anstrengendes Flimmern wahrnimmt.

In welchem Modus man auch arbeitet, Vizawrite meldet sich in jedem Fall nach relativ kurzer Ladezeit auf seinem eigenen Screen und einem geöffneten Textfenster, das wie jedes herkömmliche Workbench-Fenster behandelt werden kann. Das ist unter anderem deshalb von Bedeutung, weil Vizawrite eine Art »Dokumenten-Multitasking« erlaubt. Das heißt, es können so viele Textfenster gleichzeitig geöffnet sein, wie es der Arbeitsspeicher und des Anwenders Sinn für Ordnung zulassen. Da Vizawrite bei Editierfunktionen wie »Ausschneiden« oder »Kopieren« vom RAM-Zwischenspeicher »Clipboard« Gebrauch macht, können Texte auch problemlos zwischen den einzelnen Fenstern ausgetauscht werden.

Einen wichtigen Teil der Textformatierung übernimmt ein stilisiertes Lineal (Bild 1), das sich am Anfang eines jeden Dokuments befindet. Leider gilt ein solches Lineal für den gesamten folgenden Text. Will man nun nur eine Überschrift mittenzentrieren und alles übrige im Blocksatz formatieren, muß man direkt nach der Überschrift ein weiteres Lineal setzen. Das gilt ebenso für jede weitere Änderung des Textaussehens.

Kleine weiße und orangefarbene Dreiecke in der Formatzeile symbolisieren die Ränder und die Tabulatoren. Ferner können sieben Symbole auf der »Oberfläche« des Lineals angeklickt werden, mit denen sich der Zeilenabstand (ein-, eineinhalb- oder zweizeilig) sowie das Textaussehen (linksbündig, mittenzentriert, rechtsbündig oder Blocksatz) bestimmen lassen.

Anders als bei einer Schreibmaschine oder den meisten übrigen Amiga-Textverarbeitungen wird die Textbreite bei Vizawrite nicht in Spalten angegeben. Das liegt vornehmlich daran, daß das Programm den eingeladenen Schriftsatz (englisch: Font) in Proportionalschrift darstellt. Weil dabei aber zum Beispiel ein »I« viel weniger Raum einnimmt als ein »W«, haben in einer Zeile mal mehr mal weniger Zeichen Platz. Die logische Konsequenz ist, daß die Text- oder Zeilenbreite nicht mehr spaltenweise, sondern als Maßeinheit angegeben werden muß. Spätestens hier nun kann Vizawrite — trotz der exzellenten Deutschkenntnisse — seine angelsächsische Herkunft nicht länger verbergen. Sämtliche Angaben zur Textformatierung müssen in Zoll angegeben werden. Die Ränder und die Tabulatoren können dadurch allerdings sehr genau positioniert werden. Dazu klickt man die entsprechenden Dreiecke an und gibt den Wert direkt ein. Zusätzlich kann man für jeden Texttabulator festlegen, ob der Text normal nach rechts weitergeschrieben, mittenzentriert oder rechtsbündig dargestellt werden soll. In den letzten beiden Fällen werden die Wörter teilweise oder grundsätzlich unter dem Tabulator nach links weitergerollt. Aufgrund eines Fehlers im Programm ist dabei aber Vorsicht geboten. Erreicht nämlich auch nur ein Buchstabe den linken Fensterrand, zieht sich der Amiga »meditierend« in seinen Schmollwinkel zurück und muß durch den unbeliebten »Dreifingergriff« neu gestartet werden.

Scrolling im Schneckentempo

Vizawrite nimmt es mit dem Prinzip des »What You See Is What You Get« (WYSIWYG) sehr genau und umbricht den Text zudem sofort bei der Eingabe, ohne daß dazu extra eine Taste gedrückt werden müßte. Mag sein, daß das der Grund für den doch langsamen Bildschirmaufbau ist. Ähnlich wie »Textcraft«, dem Urahn aller Amiga-Textsysteme, rollt das Geschriebene bei Vizawrite so langsam über die Mattscheibe, als bewege es sich durch zähflüssigen Honig. Das gilt für das horizontale noch viel mehr als für das vertikale Scrolling, das dem Anwender aber auch schon ein gehöriges Maß an Geduld abverlangt.

Manche Programme (wie etwa Textomat Amiga) bieten eine automatische Silbentrennung, die ein allzu starkes »Ausfransen« des Randes verhindert, in dem sie blitzschnell überprüft, ob ein nicht mehr in die Zeile passendes Wort getrennt werden kann. Auf diesen Komfort muß man bei Vizawrite leider verzichten. Es besteht lediglich die Möglichkeit, Trennvorschläge zu setzen, die gegebenenfalls berücksichtigt werden. Allerdings arbeitet diese Option, bedingt durch einen Programmierfehler, nicht mit einem rechtsbündigen oder blockgesetzten Format zusammen, weil der vom Computer gesetzte Trennstrich am rechten Rand übersteht und beim ausgedruckten Text gar nicht mehr auftaucht.

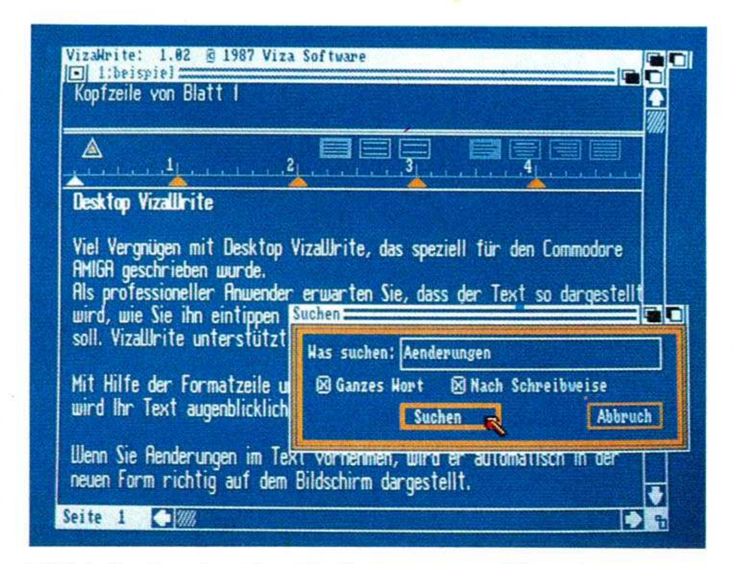


Bild 1. Suchgadget im Arbeitsfenster von Vizawrite

Auch die Funktionstasten können nicht vom Anwender belegt werden. Das macht Vizawrite aber weitgehend durch den (etwas umständlicheren) Einsatz von sogenannten Textbausteinen wieder wett. Mit ihnen lassen sich dann sogenannte Floskeldateien anlegen. Ein nettes Detail bietet das Programm mit dem Menüpunkt »Information«. Wird er angewählt, erhält der Anwender eine Textstatistik, die haarklein aufschlüsselt, wie viele Worte, Sätze und Absätze das Dokument enthält. Weitaus praktischer ist allerdings die Angabe über den verbleibenden freien Textspeicher sowie die Möglichkeit, einen Text durch ein Paßwort vor dem Zugriff Unbefugter zu schützen. aber brauchbare Ansätze in dieser Hinsicht sind durchaus erkennbar.

Auf den Spuren Gutenbergs

So ist es zum Beispiel möglich, den Text mit sämtlichen Schriftarten zu gestalten, die der Amiga beherrscht. Im Lieferumfang enthalten sind zwar nur die Workbench-Fonts, aber auch selbsterstellte oder gekaufte Schriften können mit Vizawrite genutzt werden. Die Möglichkeiten der Textgestaltung, die sich daraus ergeben, sind enorm, zumal das Programm den Zeilenabstand bei Verwendung unterschiedlich großer (also hoher) Schriften selbständig ausgleicht. Fast

Bild 2. Vizawrite fügt Grafiken in das Dokument ein

Bei der eigentlichen Textbearbeitung — Blöcke definieren, löschen, kopieren und so weiter — fällt Vizawrite weder positiv noch negativ aus dem Rahmen. Sehr erfreulich ist hingegen, daß Vizawrite beliebig lange (also auch mehrzeilige) Kopf- und Fußzeilen gestattet, die als »Fixtext« auf jeder Seite ausgedruckt und getreu dem WYSIWYG-Motto bereits auf dem Bildschirm automatisch eingeblendet werden.

Will man Vizawrite bei seinem vollen Namen nennen, so kommt man um den Ausdruck »Desktop« nicht umhin. Das bedeutet, wörtlich übersetzt, zwar schlicht »Schreibtischoberfläche«. Die Marketingstrategen zielten aber damit wohl ganz bewußt auf das Unterbewußtsein des Lesers, bei dem Assoziationen an den Begriff »Desktop Publishing« (DTP) geweckt werden sollen. Davon kann zwar auch mit viel Wohlwollen keine Rede sein, noch interessanter mag für manchen Anwender eine Funktion zur Einbindung von Grafiken (Bild 2) sein, mit der es zum Beispiel möglich wird, in einen Geschäftsbrief Balkendiagramme zu integrieren. Die Bilder können nach dem Laden noch verschoben oder sogar in ihrer Größe verändert werden. Dieser Flexibilität stehen leider zwei Einschränkungen gegenüber: Zum einen sind nur zweifarbige IFF-Bilder verwendbar, zum anderen kann Text nicht neben einer Grafik plaziert werden.

Irgendwann kommt der Zeitpunkt, an dem der Anwender sein Werk gern schwarz auf weiß in Händen halten möchte. Beim Ausdruck gibt sich Vizawrite aber wählerisch und weigert sich schlicht, mit »Exoten« zu kommunizieren. Mit anderen Worten: Ein Drucker, der weder zu einem der Vizawrite- noch zu den Preferences-Treibern kompatibel ist, kann nicht oder nur unzureichend mit dem Programm zusammenarbeiten. Wünschenswert (und eigentlich selbstverständlich) wäre die Möglichkeit einer individuellen Druckeranpassung, wie sie etwa »Textomat« oder »UBM Text V2.2« bieten.

Aber auch ohne diese Hürde ist es nicht leicht, auf Anhieb einen zufriedenstellenden Ausdruck zu erhalten. Hat man zum Beispiel mehrere Fonts kombiniert und gibt den Text nun in der Pica-Schriftart des Druckers aus, entspricht das Ergebnis nur sehr selten dem Aussehen auf dem Bildschirm. Das liegt daran, daß nicht jede Schrift mit jedem Zeichenabstand des Druckers kombiniert werden kann. Nachteilig wirkt sich das vor allem bei vielen 9-Nadel-Geräten aus, die Text in Schönschrift oftmals nur in Verbindung mit einem festen Abstand von zehn Zeichen pro Zoll (CPI) ausgeben können.

Wer seine Texte dennoch in NLQ zu Papier bringen will, muß — laut Handbuch — einen Font mit einer Größe von 12 oder mehr Pixeln Höhe verwenden. Bei unserem Test erwies sich diese Angabe jedoch als nicht allgemeingültig. Einzig Emerald 17, Emerald 20 und Ruby 15 gaben einen blockgesetzten Text auf dem Drucker originalgetreu wieder, wenn mit 10 CPI gedruckt wurde. Alle anderen Fonts arbeiteten nur mit Schmal- (15 CPI) und Proportionalschrift einwandfrei zusammen; manche »vertrugen« sich auch noch mit Elite (12 CPI). Bleibt die Möglichkeit, das Dokument als Hardcopy auszugeben. Das Ergebnis vermag aber nur auf einem 24-Nadel-Drucker zu überzeugen.

Ein positiver Aspekt an diesem Programmteil ist die Möglichkeit, den Ausdruck jederzeit unterbrechen oder beenden zu können. Der Pufferspeicher des Druckers wird allerdings nicht immer gelöscht, so daß bei einem erneuten Andruck hin und wieder Reste vom vorhergehenden Versuch mit ausgegeben werden.

Letztlich muß man sich fragen, warum es so lange gedauert hat, Vizawrite fertigzustellen. Das Programm bietet in keinem Bereich mehr als vergleichbare, zu einem früheren Zeitpunkt erschienene, Produkte. In manchen Punkten schneidet das Programm sogar schlechter ab als seine Mitbewerber. Das gilt vor allem für die Geschwindigkeit, die daran zweifeln läßt, ob Vizawrite tatsächlich zu einhundert Prozent in Maschinensprache geschrieben wurde. Es sind in erster Linie vier Punkte, die das Gesamtbild etwas aufzuhellen vermögen: die Fähigkeit, mehrere Textfenster gleichzeitig zu öffnen, die mit leichten Abstrichen flexible Grafikeinbindung und die Möglichkeit, beliebig viele verschiedene Fonts in Proportionalschrift darzustellen und auch auszudrucken sowie die Serienbrieffunktion.

Demgegenüber stehen die spartanische, völlig unzureichende Druckeransteuerung und -anpassung sowie diverse kleine Mängel im Bereich der Bedienungsfreundlichkeit.

Unverständlich ist auch, wie einige zum Teil eklatante Fehler im Programm, die schleunigst beseitigt werden sollten, dem Hersteller unbemerkt bleiben konnten. Fairerweise bietet DTM einen Update-Service für registrierte Vizawrite-Käufer an. Das werden wohl vor allem jene sein, die einen 24-Nadel- oder gar Laserdrucker besitzen, sich an der Geschwindigkeit von Vizawrite nicht stören und Wert auf eine gute Optik ihrer Texte legen.

(Karsten Lemm/pa)

AMIGA-WERTUNG

Software: VizaWrite

6,9 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■			
Dokumentation	■	■	■	■		
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■	■	
Leistung	■	■	■			

Fazit: Vizawrite ist interessant für Anwendungen, bei denen es auf abwechslungsreiche Gestaltung des Schriftbildes oder die Einbindung zweifarbiger Grafiken ankommt. Für eine Textverarbeitung, die diese Funktion nicht benötigt, ist die Verarbeitungsgeschwindigkeit zu langsam.

Positiv: Mehrfensterverarbeitung; vielfältige Funktionen zur Textgestaltung; Serienbrieffunktion; kein Kopierschutz, Update-Service.

Negativ: Sehr langsam; Programmierfehler führen zum Absturz; keine individuelle Druckeranpassung; Schwierigkeiten beim Ausdruck; keine Trennautomatik.

DATEN

Produkt: Vizawrite
Preis: 198 Mark
Hersteller: Viza Software Limited
Anbieter: DTM, Poststr. 25, 6200 Wiesbaden, Tel. 0621/560085

Grafikillusionen wie im Zeichentrickfilm

AMIGA test

Der »Animator: Apprentice« von Martin Hash ist eine Premiere: Zum erstenmal berechnet der Amiga Zeichentrickfiguren à la Disney. Die Ergebnisse sind erstaunlich, doch der Weg zur fertigen Figur ist trotz guten Hilfsprogrammen lang.

Animator: Apprentice (»Lehrling«) zeichnet sich vor allem dadurch aus, daß eine Zeichentrickfigur aus vielen Gliedern zusammengesetzt wird, die alle unterschiedliche Bewegungen ausführen können und mit einem Editor dreidimensional erstellt werden. Um die Figuren möglichst natürlich wirken zu lassen, arbeitet das Programm hauptsächlich mit gebogenen Flächen, um beispielsweise ein Gesicht darzustellen. Alle bisher erhältlichen Grafikprogramme haben mit Krümmungen große Schwierigkeiten. Zusätzlich werden die einzelnen Flächen mit natürlichen Details sozusagen beklebt, die jeweils mit einem Malprogramm wie Deluxe-Paint erstellt wurden und vom Programm automatisch auf die dreidimensionale Figur aufgesetzt werden. Um dann möglichst realistische Aufnahmen der Animationen mit einem Videorecorder zu ermöglichen, kann das Programm den Bildschirmrahmen auch ganz abschalten.

Da all diese Funktionen jedoch sehr speicherplatzfressend sind, benötigt das Programm als Minimum einen Hauptspeicher von 1 MByte. Bei umfangreicheren Animationen ist ein noch größerer Hauptspeicher zu empfehlen.

Alle Teile sind ein Ganzes

Damit die einzelnen Bilder ohne ständiges Diskettenwechseln aufgenommen werden können, sind zwei Diskettenlaufwerke notwendig.

Um nun einen Zeichentrickfim mit Animator: Apprentice zu erstellen, muß man vor allem die einzelnen Figuren genau festlegen. Da sich alle Figuren aus mehr oder weniger vielen Gliedmaßen zusammensetzen, benötigt das Programm von jedem einzelnen Glied einer Figur ein dreidimensionales Bild. Es setzt sich aus mehreren Teilen zusammen: Der Vorderansicht, der Seitenansicht und den Oberflächendetails der Figur.

Alle diese Ansichten werden mit Hilfe von Deluxe-Paint erstellt. Dazu sollte man sich allerdings vor der Grafikeingabe genau überlegen, wie groß welches Glied der Figur sein soll, da sich das später kaum noch ändern läßt. Die Vorderansicht und die Seitenansicht werden dabei lediglich als einfarbige Silhouetten gezeichnet; sie dienen dann dem Animator: Apprentice dazu, das gewünschte Glied dreidimensional darzustellen.

Um diese Dreidimensionalität zu erreichen, muß als erster Apprentice-Programmteil »Sculpt« (Bild 1) geladen werden. Dieses Programm entwirft nun aus den Silhouetten der Vorder- und Seitenansicht einen dreidimensionalen Körper. Dazu wird der Körper in dünne waagerechte Scheiben aufgeteilt, deren Konturen vom Programm automatisch gezeichnet werden, was aber bei etwas komplizierteren Gliedmaßen nicht immer zu einem befriedigenden Ergebnis führt. Deshalb können die Scheiben auch mit der Maus erstellt und korrigiert werden. Ist der Körper mit allen Scheiben im Computer gespeichert, so werden die vorher mit Deluxe-Paint erstellten Oberflächendetails automatisch »aufgeklebt«. Das Arbeiten mit »Sculpt« erfordert zwar etwas Übung, wird aber durch eine gute Menüführung erleichtert. Ihr fehlt allerdings eine Sicherheitsabfrage bei »Quit«, so daß man leicht einmal das Programm verläßt, statt ein Objekt zu speichern, da die Menüpunkte »Save« und »Quit« direkt übereinanderliegen. Dieser Fehler ist auch in den anderen Programmteilen des Apprentice vorhanden.

Die Silhouetten sowie die Oberflächen der Körper lassen sich auch sehr gut mit Hilfe eines Videodigitizers (DigiView) erstellen, was in der Anleitung auch ausführlich erklärt wird.

Verkettete Bewegungen

Ist die Form der Gliedmaßen einer Figur definiert, muß der Computer natürlich auch ihre Stellung innerhalb der Figur kennen. Zusätzlich ist wichtig, daß der Computer über die Abhängigkeiten der Gliedmaßen untereinander Bescheid weiß.

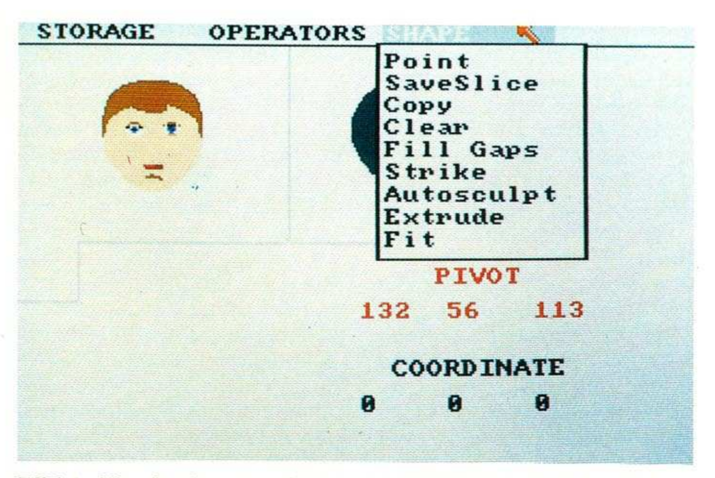


Bild 1. Mit »Sculpt« werden dreidimensionale Objekte mit beliebigen Oberflächen entworfen

Soll zum Beispiel eine menschliche Figur in Bewegung gebracht werden, so ist es für die Animation wichtig, daß sich der Unterarm und die Hand zwangsläufig mitbewegen, wenn der Oberarm oder die Schulter bewegt werden.

Mit einem weiteren Programmodul werden nun Stellung und Abhängigkeit der Gliedmaßen voneinander eingegeben. Dabei stellt der Apprentice die einzelnen Gliedmaßen als Striche auf dem Bildschirm dar, was für eine hohe Arbeitsgeschwindigkeit sorgt. Die Abhängigkeiten der Gliedmaßen untereinander werden als Schema auf dem Bildschirm angezeigt. Für jedes Glied innerhalb dieses Schemas gibt man mit der Maus die Stellung innerhalb der Figur ein.

Ist die eigentliche Figur erstellt, braucht der Computer natürlich noch genaue Anweisungen, wie die einzelnen Gliedmaßen bewegt werden sollen, wobei die vorher angesprochenen Abhängigkeiten natürlich eine große Rolle spielen. Diese Anweisungen erhält er im Programmteil »Action« (Bild 2). Hier werden für jedes einzelne Glied der Figur (entweder mit der Maus oder als Zahlenwerte) die Bewegungen um die drei Achsen sowie Verschiebungen eingegeben. Dabei wird jeder Bewegung eine bestimmte Bildzahl zugewiesen. Dabei ist es zusätzlich möglich, am Anfang und Ende der Bewegung eine Beschleunigung oder Abbremsung des Gliedes einzubauen, wie das im »richtigen Leben« ja auch vorkommt. Sind alle Bewegungen eingegeben, kann man sich die Figur bereits als »Strichmännchen« aus jeder beliebigen Richtung (vorne, hinten, rechts, links, oben und unten) ansehen.

Das Erlernen der Arbeit mit dem »Action«-Editor dauert relativ lange, da die Vielfalt der vorhandenen Menüpunkte und Funktionen verwirrend ist. Die Anleitung behandelt diesen Punkt leider nur ziemlich knapp, so daß es einige Zeit dauern wird, bis man jede vorhandene Funktion auch beherrscht. Es ist außerdem oft schwierig, natürliche Bewe-

gungsabläufe für den Computer umzusetzen, wenn man nicht schon ein guter »Animateur« ist.

Im »Action«-Programmteil selbst steckt ein kleiner Programmfehler: das Unterverzeichnis »act« wird nicht automatisch auf der Datendiskette angelegt, was bis zum Absturz des Computers führen kann. Dieses Problem läßt sich aber leicht beheben, wenn im CLI

```
makedir data:act
```

eingegeben wird. Mit »data« ist natürlich die eigene (nicht die mitgelieferte!) Datendiskette gemeint.

Ist die Bewegungsart der verwendeten Figuren definiert, so benötigt das Programm nur noch ihre Bewegungen auf einer imaginären »Bühne«, die auf dem Bildschirm von oben zu sehen ist (Bild 3). Die Bewegung einer Figur ist dabei mit Kreisen gekennzeichnet, die durch Linien miteinander verbunden sind. Die in den Kreisen zu sehenden Pfeile stellen die Bewegungsrichtung der jeweiligen Figur dar. Im Bild wurden mehrere Figuren auf die »Bühne« gebracht, was an den verschiedenen Linien zu erkennen ist.

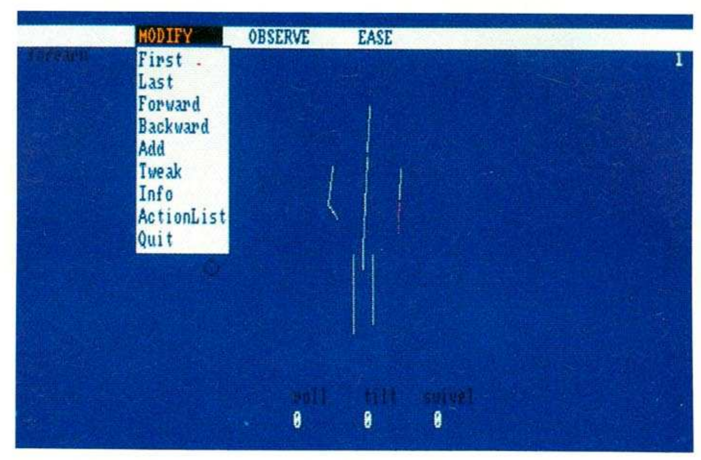


Bild 2. Die Bewegungen der Einzelteile einer Figur werden in »Action« auf einfache Weise festgelegt

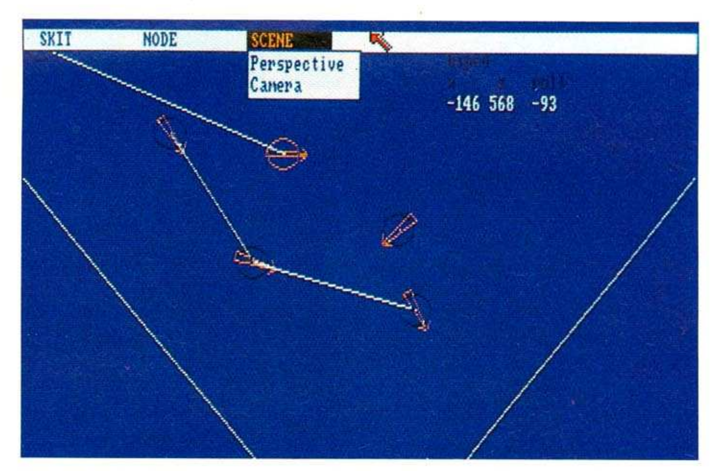


Bild 3. Der Ablauf der kompletten Szene wird auf einer Bühne konstruiert, die dem Blickfeld angepaßt ist.

Action auf der Bühne

Im »Director« (das ist der Name des Programmteils, der die »Bühne« verwaltet) gibt man die Bewegungen der einzelnen Darsteller sehr komfortabel über die Maus ein (die Koordinaten lassen sich auch über die Tastatur eintippen). Die Bewegung der »Kamera« ist ebenso programmierbar wie die Bewegung der Figuren.

Zusätzlich lassen sich noch die Position der Lichtquelle, die Beleuchtung der Bühne (diffuses oder eher gerichtetes Licht) und die Hintergrundfarbe des Bildschirms einstellen.

Nun könnte die endgültige Animation eigentlich bereits berechnet werden. Da das aber relativ lange dauert, gibt es zusätzlich noch ein Probe-Programm (Rehearsal), das alle bisher gemachten Eingaben verwendet (Figuren- und Kamerabewegungen auf der Bühne stimmen also bereits mit der endgültigen Animation überein) und lediglich die Figuren werden noch als »Strichmännchen« dargestellt. Die Rechengeschwindigkeit ist dabei sehr hoch, da ja die aufwendigen Schattierungen und dreidimensionalen Berechnungen wegfallen.

Ist die Animation nun endlich perfekt, wird sie mit dem »Record«-Programm auf Diskette aufgenommen. Das Programm kann das in zwei verschiedenen Formaten tun: gepackt oder im IFF-Format. Wobei aber nur gepackte Bilder in »Echtzeit« abspielbar sind. Gemeinsam haben die beiden Formate, daß die einzelnen Bilder der Animation auch als einzelne Dateien auf Diskette abgelegt werden, was sie zwar leicht zu bearbeiten, aber langsam zu laden macht. Besonders interessant ist natürlich die automatische Schattierung der animierten Figuren, die die vorher definierte Position und Art der Lichtquelle berücksichtigt. Sie kann mit 2, 4, 8, 16 oder 32 Farben oder auch im Hold-and-Modify-Modus erfolgen. Die Aufnahme der Animation dauert pro Bild etwa zehn Minuten, kann aber völlig unbeaufsichtigt — etwa in der Nacht, wenn sowieso keiner am Computer sitzt — stattfinden. Auf einer Diskette bringt der Apprentice etwa 120 (gepackte) Bilder unter.

Die von »Record« erstellte Diskettendatei wird nun mit Hilfe des »Display«-Programmes in »Echtzeit« auf dem Bildschirm angezeigt (Bild 4). Der Zeitunterschied zwischen den Bildern ist bei etwas aufwendigeren Animationen allerdings deutlich zu erkennen. Da die Anzeige auch jederzeit angehalten werden kann, lassen sich die Bilder auf einem Einzelschritt-Videorecorder aufnehmen, was für fließende Bewegungen sorgt.

Fehlender Hintergrund

Bei der Aufnahme mit einem Videorecorder tritt ein Problem auf, das allen amerikanischen Programmen auf deutschen Amiga-Modellen zu schaffen macht: Durch die höhere Zeilenzahl der deutschen Geräte bleibt (auch bei Abschalten des Bildschirmrahmens) ein

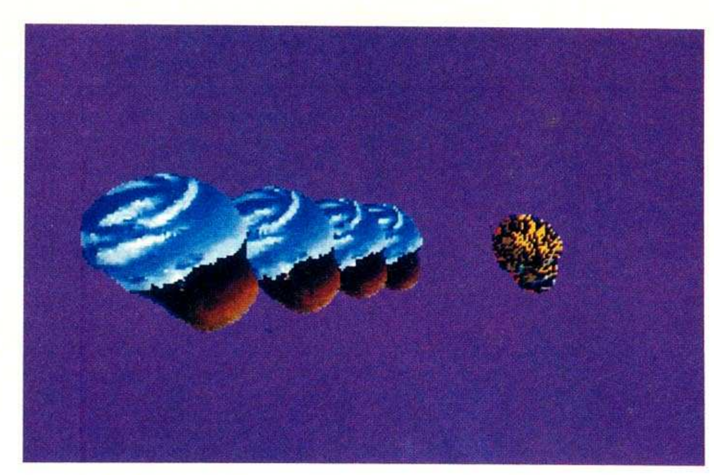


Bild 4. In der fertigen Animation bewegen sich die Figuren butterweich über den Bildschirm

unbenutzter Streifen am unteren Rand des Bildschirms, der natürlich auch auf Video aufgenommen wird. Bei der Anzeige des Videofilms über ein normales Fernsehgerät wird der Streifen dann natürlich auch sichtbar.

Eines fehlt beim Apprentice völlig: Ein Hintergrund für die animierten Figuren. Es ist nicht möglich, ein IFF-Bild als Hintergrund zu verwenden (außer man definiert es ebenfalls als Figur und verbraucht dadurch zusätzlich Speicher). In der Anleitung wird zu diesem Thema lapidar empfohlen, sich mit einem Genlock-Interface (Gerät zur Einblendung eines Videobildes auf den Amiga-Bildschirm) zu behelfen. Wer allerdings nicht zusätzliches Geld für dieses Gerät ausgeben will, kann höchstens noch mit Hilfe eines IFF-Grafikprogrammes die Einzelbilder der Animation mit einem Hintergrund versehen. Dann kann die Animation aber nicht mehr in Echtzeit angesehen, sondern nur noch mit dem Videorecorder »zum Leben erweckt« werden. Das englische Handbuch des Apprentice ist sehr übersichtlich aufgebaut und mit Hardcopies des Programms gut illustriert, teilweise aber etwas zu knapp gehalten. Hier hilft auch die kurze schrittweise Einführung im Anhang nicht weiter, weil dort ständig auf das Handbuch verwiesen wird. Die oft verwendeten englischen Ausdrücke für Körperteile wie Becken oder Oberschenkel dürften auch nicht jedermann bekannt sein. Martin Hash, der Programmierer des Apprentice, ist ständig daran, das Programm weiter zu verbessern. Eingetragene Benutzer kommen in den Genuß von regelmäßigen Updates und »Newslettern«, die wichtige Techniken und Tricks erläutern. Inzwischen gibt es auch einige Zusatzdisketten, die fertige Objekte wie zum Beispiel verschiedene Buchstaben und Schriften enthalten.

Insgesamt ist Animator: Apprentice ein Programm, das von Funktionenvielfalt wie Preis (DM 595 Mark) her eher für den »Profi« im Bereich der Computeranimation bestimmt ist. Interessant wird das Programm erst, wenn es mit Speichererweiterung, Genlock-Interface und Videorecorder voll ausgenützt wird. (Andreas Lietz/jk)

AMIGA-WERTUNG

Software: Animator: Apprentice

7,4 von 12	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■				
Dokumentation	■	■	■	■		
Bedienung	■	■	■	■	■	
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Der Apprentice ist ein leistungsfähiges Animationsprogramm für »lebensnahe« Figuren. Die Funktionenvielfalt kann jedoch nur mit entsprechenden Zusatzgeräten voll ausgenutzt werden.

Positiv: Durch Schattierung sehr realistische Darstellung; guter 3D-Editor; viele Funktionen zur Gestaltung der Animation.

Negativ: Keine Sicherheitsabfrage bei »Quit«; Hintergrund kann nicht bearbeitet werden; Aufteilung in einzeln zu ladende Programme.

DATEN

Produkt: Animator: Apprentice
Preis: 595 Mark
Hersteller: Hash Enterprises
Anbieter: Intelligent Memory, Basaltstr. 58, 6000 Frankfurt, Tel. 069/7071102

Klangzauber aus Bits und Bytes

AMIGA test

Mit Hilfe eines kleinen Zusatzgerätes ist jeder Amiga-Besitzer in der Lage, seine eigenen Sounds zu digitalisieren. Der »DeLuxe Sound Digitizer« bietet dies zu einem angemessenen Preis.

Fast jedes Spiel bedient sich digitalisierter Klänge, um noch realistischer zu wirken. Aber auch Musikprogramme wie Sonix und das Deluxe Music Construction Set bieten Naturklänge zur Auswahl. Die Soundfähigkeiten des Amiga basieren auf einer relativ simplen Technik. Vier Digital-Analog-Wandler machen die Bits und Bytes, die im Speicher des Amiga stehen, hörbar. Natürlich müssen die Bytes in einer bestimmten logischen Reihe zueinander stehen, ansonsten würde der Amiga nämlich nur ein Chaos aus Rauschen und Brummen von sich geben. Die Aufgabe eines Digitizers oder Samplers ist es, einen Analogklang für den Computer zu übersetzen. Bei diesem Vorgang tastet der Digitizer das Analogsignal mit einer bestimmten Rate ab (Sampling-Rate) und schickt die gewonnenen Werte in den Speicher des Computers. Die Qualität des digitalisierten Klanges hängt dabei maßgeblich von der Samplingrate und der Auflösung der A-D-Wandler beziehungsweise D-A-Wandler ab. Die D-A-Wandler im Amiga arbeiten mit einer Auflösung von 8 Bit, was einen guten Kompromiß zwischen Klangqualität und dafür benötigter Speicherkapazität darstellt. Daß sich damit natürlich keine Klänge in CD-Qualität (16 Bit) erreichen lassen, ist verständlich.

Dennoch lassen sich Sounds und Instrumente in sehr guter Qualität aufzeichnen, wie der DeLuxe Sound Audio-Digitizer von Hagenau Computer beweist (Bild 1). Dieser Hardware-Zusatz wird an den Parallelport des Amiga gesteckt und belegt zusätzlich noch den seriellen Port (aufgrund der Spannungsversorgung). Über einen in das Modul integrierten Lautstärkeregler lassen sich alle gängigen Tonquellen wie Mikrofone oder CD-Player anpassen. Leider sind weder der parallele noch der serielle Port durchgeführt. DFÜ-Anhänger und Druckerbesitzer müssen somit ständig umstecken. Um einen Klang optimal digitalisieren zu können, müssen ein paar Vorbereitungen getroffen werden. Beim Anpassen des Lautstärkepegels ist die Software sehr hilfreich, da sie eine direkte Ausgabe des Klangmaterials über den Digitizer erlaubt (Bild 2). Der Benutzer hört damit eventuelle Verzerrungen sofort. Als zusätzliche Hilfe flackert der Bildschirm des Amiga bei Spitzenpegeln rot

UMSTEIGEN HEISST AUFSTEIGEN



Jahrelang haben Sie mit dem Commodore 64 gearbeitet. Doch die Ansprüche steigen. Nun endlich – der Traum ist erfüllt: Der Amiga steht auf Ihrem Schreibtisch. Doch wie geht's weiter? Wieder ganz von vorn anfangen? Läßt sich das mittlerweile erworbene Know-how nicht auch für den neuen Rechner nutzen? Wie läßt sich die bestehende Hard- und Software auch weiterhin einsetzen? Typische Umsteigerprobleme, zu denen es verblüffend einfache Lösungen gibt. Mit dem Aufsteigerbuch. Hier finden Sie Schritt für Schritt einen schnellen Einstieg in die Workbench, lernen, wie Sie das CLI einsetzen und die Multitaskingfähigkeit des Amigas nutzen können. Alles immer im Vergleich zu Ihrem alten Rechner. So werden beispielsweise detailliert die Unterschiede des AmigaBASIC zum vertrauten BASIC V2.0 herausgestellt. Die beste Voraussetzung, um schnell eigene BASIC-Programme auf dem Amiga entwickeln zu können. Auch andere, bereits vorhandene Sprachen lassen sich sinnvoll nutzen – ob C oder 68000-Assembler. Das Aufsteigerbuch führt Sie schnell und effektiv in diese Sprachen ein, so daß Sie sich für die richtige entscheiden können. Und die bange Frage nach der weiteren Verwendbarkeit der alten Hard- und Software? Auch die wird hier beantwortet. Monitor und Drucker lassen sich möglicherweise genauso problemlos einsetzen wie der gute alte C64 selbst, z. B. als 60 Kbyte großen Druckerpuffer! Für den eigentlichen Daten-Transfer von Texten, Bildern, Dateien und BASIC-Programmen enthält das Buch fertige Konvertierungsprogramme, die soweit möglich auch die Anpassung der übertragenen C64-Daten bzw. -Programme an den Amiga übernehmen. Können Sie sich einen leichteren Umstieg auf den Amiga vorstellen?

Amiga Aufsteigerbuch
320 Seiten, DM 39,–

Gleich richtig aufsteigen!

Mit der richtigen Zeitschrift: DATA WELT. Hier finden Sie alles rund um den Amiga: kreative Projekte, Software-Tests, Interviews und wichtige News. Aber auch, was sich sonst so in der Computerszene tut, bleibt Ihnen nicht verborgen. DATA WELT – das aktuelle Computermagazin.

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
zzgl. DM 5,- Versandkosten
unabhängig von der bestellten Stückzahl
per Nachnahme ☐ Verrechnungsscheck liegt bei

auf. Nach Anpassung der Lautstärke sollte man sich als nächstes Gedanken über die Sampling-Rate machen, die von 5637 bis 28867 Hz reicht. Je höher diese Aufzeichnungsfrequenz ist, desto besser ist auch die Wiedergabequalität eines Klanges. Allerdings resultiert daraus proportional auch ein höherer Speicherbedarf. Die Gelegenheit das Klangmaterial sofort zu hören (Direct Output), erleichtert die Wahl der richtigen Digitalisierungsrate enorm. Die eigentliche Aufnahme kann nun mit Hilfe der Maus (Start anklicken) oder automatisch ab Erreichen eines bestimmten Lautstärke-Levels (Threshold) beginnen. Leider läßt sich dieser Level vom Benutzer nicht einstellen. Während des Sample-Vorgangs kann der Klang mitgehört werden. Die Aufnahme kann jederzeit durch einen Mausklick beendet werden.

Gute Soundqualität

Mit dem digitalisierten Klang lassen sich interessante Verfremdungen durchführen. So lassen sich Klänge in ihrer Amplitude (Lautstärke) oder Abspielfrequenz (Tonhöhe) modulieren.

Die Wellenformen, mit denen das Klangmaterial beeinflußt wird, sind dabei mit der Maus frei definierbar. Die Amplituden-Modulation läßt sich vorwiegend für Tremolo- oder Verzerrungseffekte einsetzen. Aus einer gesampelten normalen Stimme läßt sich auf diese Weise eine Roboterstimme gewinnen. Mit der Frequenzmodulation lassen sich extreme Tonhöhenschwankungen oder auch Phasing-Effekte erzielen. Bei den meisten Sounds kann die ganze Palette verschiedener Klangbeeinflussungen während des Abspielvorgangs in Anspruch genommen werden.

Es macht Spaß mit diesen verschiedenen Modulationen zu experimentieren, dennoch sind die erzielbaren Ergebnisse klanglich nicht besonders verwertbar. Sehr nützlich dagegen sind die verschiedenen Speicherformate, die das Programm anbietet. Das Dump-Format, das auch der bekannte Futuresound-Digitizer verwendet, bildet die einfachste Möglichkeit, einen Klang zu speichern. Gerade bei längeren Musikstücken hat sich dieser Standard deutlich durchgesetzt. Natürlich kommt auch der IFF-Standard nicht zu kurz. Wer sich neue Instrumente für das Deluxe Music Construction Set aufzeichnen will, steht vor keinem Problem. Auch für das etwas vom IFF-Standard abweichende Sonix lassen sich mühelos Instrumente kreieren. Vor dem Speichern kann festgelegt werden, ob es sich um einen normalen oder um einen geloopten (sich ständig wiederholenden) Sound handelt. Leider ist das Setzen der Wiederholschleife bei Klängen nicht besonders gut gelöst. Mit einem einzigen Slider muß mühevoll ein passender Punkt gefunden werden, ab dem sich das Sample wiederholen soll. Eine grafische Aufbereitung des Klangmaterials würde bei der Suche nach einem geeigneten Loop-Point sehr von Nutzen sein. Doch dafür entschädigt so manche andere Funktion. Für Experimentierfreudige stellt das Programm einen Echoeffekt zur Verfügung. Mit der Maus sind bis zu fünf Wiederholungen einstellbar. Über einen Slider ist dabei noch die Zeit für die jeweilige Echodauer einstellbar. Echo ist in diesem Programmteil ein irreführender Begriff. In Wirklichkeit wird jedes Echo eines Klanges grundsätzlich vom Samplebeginn weg gestartet. Dieser Tatsache ist es auch zuzuschreiben, daß sich der bekannte »Nineteen-Effekt« (Staccatoartige Wiederholungen) relativ leicht realisieren läßt.

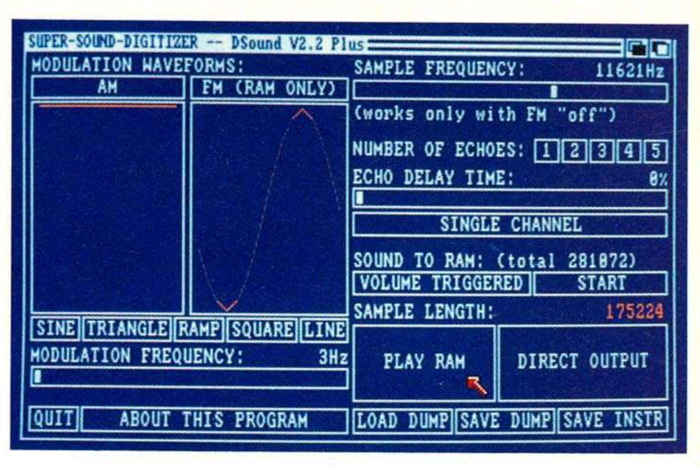


Bild 2. Das Hauptmenü des DeLuxe Digitizers mit seinen grafisch dargestellten Modulationsmöglichkeiten

Zusatzqualifikationen

Auf der Diskette befinden sich neben dem Digitizer-Programm selbst noch einige nützliche Befehle. Mit Hilfe dieser kleinen Programme lassen sich Musikstücke direkt auf Diskette digitalisieren oder zu einem IFF-Bild gesampelte Klänge abspielen. Die in Assembler geschriebene Software ist mit keinem Kopierschutz versehen, somit kommt auch die wachsende Schar von Festplattenbesitzern auf ihre Kosten.

Amiga-Anwender, die sich mit den Soundmöglichkeiten dieses Computers auseinandersetzen wollen, stellen die Hauptzielgruppe dieses Hardware-Zusatzes dar. Besonders Besitzer des Deluxe Music Construction Sets oder von Sonix sollten dieses Programm einmal näher ins Auge fassen, da ein Digitizer die Klangvielfalt dieser Musikprogramme enorm erweitern kann.

Einige wichtige Neuerungen sind als Update für die Software in Planung oder sollen laut Hersteller bis zum erscheinen dieses Testes noch in eine neue Version eingearbeitet sein. Die Leserroutinen des Record-Befehls sollen verbessert, sowie auch eine Vorhör-Möglichkeit integriert werden. Weiterhin soll die Abspielgeschwindigkeit der Samples über die Maustasten kontrollierbar sein.

Die Hardware des DeLuxe Sound Digitizer kann auch direkt mit anderen Programmen, die Samples erstellen, zusammenarbeiten. Laut Hersteller kann die Software von »Futuresound«, »Perfect Sound«, »Aegis Audio Master« und dem Digitizer unserer Schwesterzeitschrift 68000er benutzt werden, um Sounds über das Modul einzulesen.

(Bernhard Carli/jk)

AMIGA-WERTUNG

Software: DeLuxe Sound Digitizer

8,4 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■				
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■	■	
Leistung	■	■	■	■		

Fazit: Für alle, die einen preiswerten Sound-Digitizer suchen, ist DeLuxe Sound genau das richtige. Sounds können in verschiedenen Musikprogrammen oder für die eigene Programmierung verwendet werden.

Positiv: In Assembler programmiert; vielfältige Speicherformate (Dump, IFF, Sonix); interessante Modulationsmöglichkeiten (AM, FM & Echo); gute Soundqualität; kein Kopierschutz.

Negativ: Loop-Point nur umständlich setzbar; Threshold nicht einstellbar; paralleler und serieller Port nicht durchgeschleift.

DATEN

Produkt: DeLuxe Sound Digitizer

Preis: 198 bis 228 Mark, je nach Ausstattung und Version

Hersteller/Anbieter: Hagenau Computer, Münsterstr. 202, 4700 Hamm, Tel. 02381/673165



Bild 1. Das komplette Paket der Amiga 1000-Version mit Software, Mikrophon und Modul

Faszination in drei Dimensionen

AMIGA test

Ein Konkurrent für das Animationsprogramm »Videoscape« macht sich stark auf dem Softwaremarkt: Mit dem halben Preis und ähnlichen Leistungen bietet »Forms in Flight« ein durchweg positives Bild.

Forms in Flight ist ein 3D-Zeichen- und Animationsprogramm, das vor allem durch seine vielfältigen Eingabefunktionen und komplexen Grafikmanipulationen verblüfft. Im ersten Teil von FIF (Forms in Flight) kann man mit einem objektorientierten Zeichenprogramm Drahtgittermodelle entwerfen. Die Ergebnisse können später mit Berechnung verdeckter Linien, Farbgebung unter Einbindung von Lichtquellen und sogar als Rot-Blau-3D-Bild dargestellt werden. Erst nach dem Zeichnen und Bearbeiten kommt Bewegung ins Bild.

Die Eingabe der Objekte kann entweder mit der Maus auf dem Bildschirm erfolgen, oder durch die Eingabe von exakten Zahlenwerten. So ist zum Beispiel bei einer Skalierung von 100 Koordinatenpunkten auf ein Pixel eine Eingabe per Maus unsinnig, wenn die Position nicht ein Vielfaches von 100 ist. Andererseits kommt das Ändern der Skalierung für den Betrachter nicht in Frage, weil das zu bildende Objekt sonst über den Bildschirm hinausgeht und nicht mehr der gewünschten Übersichtlichkeit genügt.

Editiert wird objektweise in hoher Bildschirmauflösung (640 x 200 oder 640 x 400 Punkte). Vor der Eingabe darf man allerdings noch auswählen, auf welcher Ebene gezeichnet wird. Bei einer direkten 3D-Eingabe bekommt man sofort einen Eindruck vom Aussehen des Objekts — allerdings, wie in der realen Welt auch, nur aus dem gegenwärtigen Blickwinkel des Betrachters. Für exaktes Zeichnen ist der 2D-Blickpunkt wohl besser (Bild 1); so kann der Grundriß eines Hauses aus der xz-Ebene gezeichnet werden, die Fassaden aus der xy- oder yz-Ebene. Zum Schluß wird das ganze Haus nur noch aus den zweidimensionalen Einzelteilen zum 3D-Objekt zusammengesetzt (Bild 2). Mit dieser Fertigbauweise werden grundsätzlich alle Grafiken zusammengestellt. Das Gesamtprodukt kann anschließend als weiteres Objekt in einer Bibliothek gespeichert werden. Aus diesen Bibliotheken können dann weitere Bilder zusammengestellt werden. »Wie erstelle ich für mein 3D-Raum diagonal liegenden Linien zur Verbindung zeichnen wollte. So ein in der Schräge des Dachs liegendes Fenster ist aus Schrägoben-Ansicht schließlich nur ein einfaches Viereck, und als solches sollte man es auch eingeben können. So wurden neben den parallel zu den Achsen liegenden Grafikebenen noch »schräge«, benutzerdefinierbare Ebenen möglich gemacht. Man gebe nur die Lage der neuen Ebene im Raum ein, und schon lassen sich schief liegende Objekte aus der simplen Senkrecht-Perspektive zeichnen. Die andere Möglichkeit wäre gewesen, das Fenster oder auch das Dach auf einer geraden Ebene zu zeichnen und dann zu drehen, um es schließlich an die richtige Stelle zu schieben. Dabei kommt man aber schnell mit schon auf dem Bildschirm vorhandenen Teilen durcheinander und arbeitet zudem lange daran, das Objekt an die richtige Stelle zu schieben. Der Benutzer kann also mit der in FIF verwendeten Methode leichter arbeiten. FIF hat außerdem sehr vielseitige Funktionen zur Objekteingabe. Betrachten wir beispielsweise die Kreation eines 12seitigen Zylinders, der im 45-Grad-Winkel zur x-Achse im Raum steht. Wer jetzt denkt, jeden Eckpunkt einzeln eingeben zu müssen, hat sich geirrt. Erst wird eine »schiefe« Ebene gewählt, auf der es die 12seitige Grundform zu entwickeln gilt. Man nehme die Option »Create-Polygon-Regular«, gebe eine 12 ein und plaziere mit der Maus das Zentrum und den Radius. Das ist alles; danach ist der Grundriß fertig. Um daraus einen Zylinder zu machen, ziehe man das Polygon einfach mit »Sweep-away« in die Länge. Wie lang der Zylinder werden soll, kann exakt eingegeben werden. Anschließend schalten wir zurück auf 3D und siehe da, der Zylinder steht schräg im Raum; genauso, wie es geplant war.

Bild 1. Große Objekte werden aus Teilen zusammengesetzt



Bild 2. Aus jedem beliebigen Winkel und variabler Entfernung darf man die Konstruktion betrachten

Sonderfunktionen im Griff

Ein Problem für andere Grafikprogramme wäre zum Beispiel, einen Aufkleber auf eine Flasche zu bringen. Mit FIF geht dies in Sekunden. Ein Viereck mit einem Ellipsenfaktor (Biegung) von 180 Grad konstruieren und dann nur noch plazieren; fertig ist die um die Flasche gelegte Grafik. Sogar Spline-Kurven sind machbar, und 2D-Objekte können mit »Surface of Revolution« in 3D-Rotationsobjekte gewandelt werden.

Durch rotierende Kopien, Spiegelungen oder Verschiebungen können auf ganz einfache Weise die kompliziertesten Grafiken erstellt werden. So kann aus einem Zylinder eine ganze Reihe von Säulen für die Akropolis werden — mit nicht mehr als vier Befehlen (einer

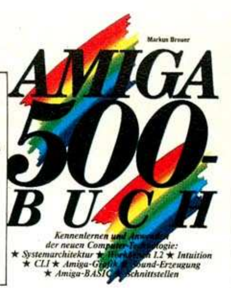
Die Amiga-Bibliothek




H.-R. Henning
Programmieren mit Amiga-Basic
1987, 360 Seiten, inkl. Diskette
Eine praxisbezogene Einführung in die Programmierung mit Amiga-Basic. Mit 100 Programmen und vielen Beispielen sowie einem Malprogramm und einer leistungsfähigen Dateiverwaltung.
Bestell-Nr. 90434
ISBN 3-89090-434-3
***DM 59,–**/sFr 54,30/öS 460,20*

NEU

P. Wollschlaeger
Amiga-Assembler-Buch
1987, 329 Seiten, inkl. Diskette
Ein 68000er-Kurs mit vielen Beispielen. Mit ausführlichem Verzeichnis aller Systemroutinen und genauer Anleitung für das Einbinden von Assembler-Routinen in Amiga-BASIC.
Bestell-Nr. 90525
ISBN 3-89090-525-0
***DM 59,–**/sFr 54,30/öS 460,20*



M. Breuer
Das Amiga 500-Buch
1987, 489 Seiten
Eine ausführliche Einführung in die Bedienung des Amiga 500. Das Handbuch dient als Nachschlagewerk beim alltäglichen Einsatz.
Bestell-Nr. 90522
ISBN 3-89090-522-6
***DM 49,–**/sFr 45,10/öS 382,20*



F. Kremser/J. Koch
Amiga Programmierhandbuch
1987, 390 Seiten, inkl. Diskette
Eine Super-Einführung in die »Interna« des Amiga: die wichtigsten Systembibliotheken, die das Betriebssystem zur Verfügung stellt, werden ausführlich anhand von Beispielen in C erklärt.
Bestell-Nr. 90491
ISBN 3-89090-491-2
***DM 69,–**/sFr 63,50/öS 538,20*

Dr. E. Huckert/F. Kremser
Amiga: C in Beispielen
1987, 237 Seiten, inkl. Diskette
Erste C-Programme, Daten und Datentypen, Operatoren, Befehle, Eingabe und Ausgabe, Arrays und Pinter, Strukturen und Verbunde, Prozeduren, der C-Präprozessor.
Bestell-Nr. 90539
ISBN 3-89090-539-0
***DM 69,–**/sFr 63,50/öS 538,20*

NEU



Bantam Books
Das Amiga-DOS-Handbuch für Amiga 500, 1000 und 2000
1987, 342 Seiten
Die Pflichtlektüre für jeden Commodore-Amiga-Anwender und Programmierer: eine Entwickler-Dokumentation zum Amiga-DOS-Betriebssystem, Version 1.2. Programmierung, interne Datenstruktur und Diskettenhandling.
Bestell-Nr. 90465
ISBN 3-89090-465-3
***DM 59,–**/sFr 54,30/öS 460,20*

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften oder in den Fachabteilungen der Warenhäuser.

Irrtümer und Änderungen vorbehalten.



Zeitschriften · Bücher
Software · Schulung

Fragen Sie bei Ihrem Buchhändler nach unserem kostenlosen Gesamtverzeichnis mit über 300 aktuellen Computerbüchern und Software. Oder fordern Sie es direkt beim Verlag an!

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0.
SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656,
ÖSTERREICH: Rudolf Lechner&Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526, Ueberreuter Media Handels- und Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0

für die Kopien einer jeden Seite). Durch Hinzufügen einiger Drei- und Vierecke wird die Akropolis perfekt. Wer will, kann die Säulen auch schief legen, denn das Verlängern von Grundobjekten geht grundsätzlich in jede Richtung.

Die fertigen Konstruktionen können fleißig manipuliert werden, wie auch das Drumherum. So kann man die Grafik im Raum rotieren lassen oder statt dessen den Blickpunkt des Betrachters um die Grafik drehen. Man kann auf die Grafik zufliegen oder denselben Effekt durch Heranzoomen erwirken. Obwohl auch dies auf den ersten Blick dasselbe zu sein scheint, liegen unterschiedliche Berechnungen zugrunde, was für weitere Bewegungs- und Animationsphasen von Bedeutung sein kann.

Um die Form eines Grafikobjektes zu ändern, kann man mit »Move node« die Ecken eines Gegenstandes herausziehen (ähnlich wie beim Aegis Animator) oder mit »Resize« die Größe verändern.

Schattierungen mit Lichtquellen

Neben diesen und vielen anderen räumlichen Manipulationen gibt es Sichtmanipulationen. Dazu gehören »Hidden Lines« und »Stereo View«. Für jede dieser Optionen (nicht nur einzeln, sondern auch gleichzeitig) genügt je ein Mausklick. »Hidden Lines« zeigt die Grafik so, wie der Betrachter sie wirklich sieht, also nur mit den Flächen, die vom Betrachtungsort tatsächlich sichtbar sind. »Stereo View« wandelt das Bild in eine Grafik für eine Rot-Blau-Brille um; je nach Tiefe der Grafik ist das Ergebnis mehr oder weniger beeindruckend (Bild 3).

Farben werden auf Flächen verteilt. Die Farbe jeder einzelnen Fläche kann sich je nach Lichtquelle ändern. Bis zu acht Lichtquellen können, mit der Richtung, in die sie scheinen, angegeben werden. Schattierungen können nach Angabe der Lichtquellen automatisch mit einberechnet oder per Hand manipuliert werden.

Animation ist ein Wort, das sehr komplexe Vorgänge umspannen kann. Stellen Sie sich vor, ein Junge sitzt auf einem Pferd in einem Kinderkarussell. Das Karussel dreht sich, das Pferd wippt auf und ab, und das Kind winkt vom Karussell aus. Das Winken ist an sich nur eine Links-/Rechtsbewegung, doch zusammen mit der Auf- und Abbewegung des Pferdes, das sich auch noch mit dem Karussell dreht, ergibt sich eine sehr komplexe Bewegungslinie.

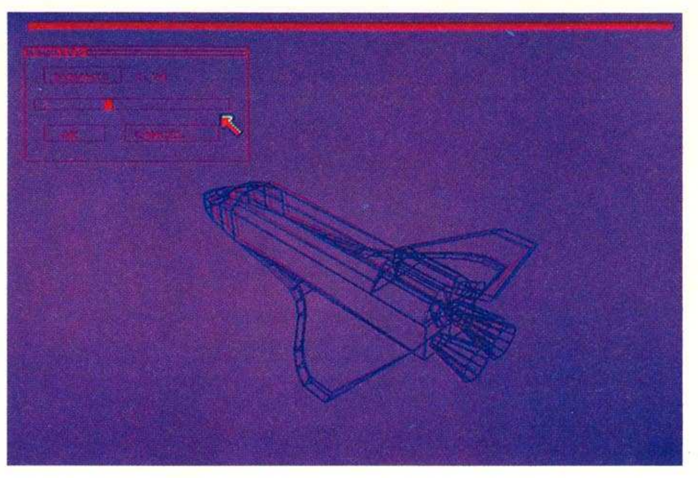


Bild 3. Das Bild ist nicht unscharf, sondern besteht aus zwei versetzten Rot-Blau-Bildern

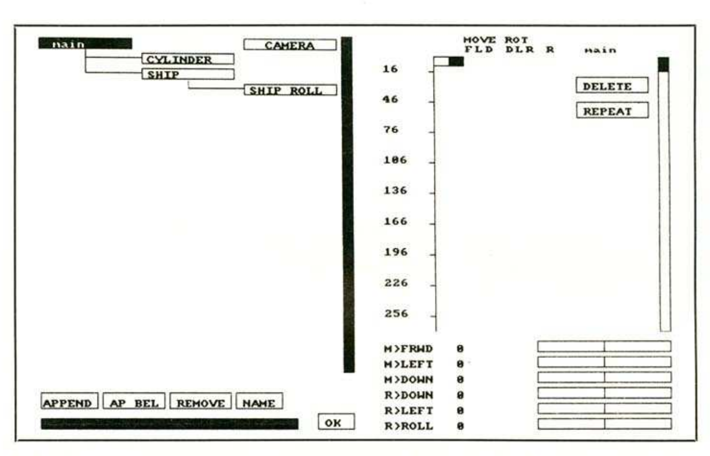


Bild 4. Die einzelnen Bewegungen lassen sich im Editor als Baumstruktur miteinander verbinden

Mit der komfortablen Bearbeitung solcher Animationen, haben sicher alle auf dem Markt befindlichen Programme Schwierigkeiten. Nicht so Forms in Flight. Mit einer Baumstruktur legt der Benutzer fest, welches Objekt an ein ein anderes gebunden ist (Bild 4). Wir erstellen also die Baumstruktur:

Karussell
    Pferd 1
    Pferd 2
        Kind
    Pferd3
    und so fort

Damit sind die drei Pferde abhängig vom Karussell. Wir geben also nur die Bewegung für das Karussell ein und die Pferde drehen sich automatisch mit. Wir geben für Pferd 1 eine Auf- und Abbewegung ein, und schon bewegt es sich auf und ab; dreht sich allerdings immer noch mit dem Hauptobjekt. Ebenso die Bewegung des Kindes. Das Programm errechnet automatisch die komplexe Bewegung im dreidimensionalen Raum.

Alle Eingaben für die Bewegung der Objekte oder der Kamera können problemlos mit der Maus durchgeführt werden und sind in kurzer Zeit gelernt.

Mit dem Handbuch wird dem Anfänger das Bedienen des Programmes so einfach wie möglich gemacht.

Zur Animation muß man allerdings auch etwas Negatives sagen: Sie ist, auch wenn fertige Bilder gespeichert werden, langsam und flimmernd. Das liegt an zwei Gründen: Zum Einen wird kein Double-Buffering angewendet (Aufbau des Bildes im Hintergrund, dann Umschalten), was das enorme Flimmern verursacht. Der andere Grund ist die Tatsache, daß jedes Bild in einer kompaktierten Version gespeichert wird. Das spart viel Speicherplatz und ermöglicht damit umfangreichere Animationen als zum Beispiel Videoscape. Vor Anzeigen einer jeden Bildsequenz muß jedoch erst einmal der Dekompressor sein Bestes tun. Im Handbuch wird davon gesprochen, daß deswegen die Erstellung eines Videos nur mit Hilfe von Einzelbildaufnahmen gute Ergebnisse bringe. Darum sei das Programm auch darauf ausgelegt, die Bilder über Nacht zu berechnen und einzeln zu speichern. Aber eine Einzelschrittoption bietet weder die Playback-Option, noch werden die Bilder in einem Standard-Format für andere Grafikprogramme gespeichert.

Forms in Flight ist also ein 3D-Draft-Programm, mit dem sich gut arbeiten läßt, das aber im Bereich Animation kein professionelles Tool darstellt. Dafür aber ist es die ideale Einführung für den Einsteiger, da alles sehr einfach gehalten wurde und mit Hilfe der Anleitung ausführlich erklärt wird.

Im Gegensatz zur amerikanischen Version wird bei der deutschen Lieferung die Rot-Blau-Brille in der Packung beigelegt, denn bei uns gibt es diese Brillen eben nicht (wie in den USA) in jedem Comic-Shop zu kaufen.

(Manfred Kohlen/jk)

AMIGA-WERTUNG

Software: Forms in Flight

8,8 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■	■	
Erlernbarkeit	■	■	■	■	■	
Leistung	■	■	■	■		

Fazit: Forms in Flight gehört mit seinen ausgezeichneten und einfach zu bedienenden Editorfunktionen in die gehobene Klasse der 3D-Draft-Programme und kann sogar als 2D-CAD-Programm verwendet werden. Die Animationsfähigkeiten sind vielseitig, aber in der gegenwärtigen Version noch nicht für den professionellen Einsatz geeignet. Einsteiger in die Computergrafik werden die gute Anleitung und Bedienung schnell schätzen lernen.

Positiv: Gutes deutschsprachiges Handbuch; 3D-Brille mitgeliefert; einfache Bedienung.

Negativ: Benötigt mindestens 1 MByte Speicher; keine Einzelbildschaltung; langsamer Bildaufbau.

DATEN

Produkt: Forms in Flight

Preis: 165 Mark

Hersteller: Micro Magic

Anbieter: Intelligent Memory, Basaltstr. 58, 6000 Frankfurt, Tel. 069/707 11 02

Dateiverwaltung zum Niedrigpreis

AMIGA test

»Qualität hat seinen Preis« — eine in der Softwarebranche gern verwendete Rechtfertigung für hohe Preise. Geht der günstige Preis für Datamat auf Kosten der Qualität? Wir haben das für Sie getestet.

Die Dateiverwaltung Datamat gibt es bereits einige Zeit auf dem Atari ST. Nun haben die Softwareentwickler von Data Becker das Programm für den Amiga umgesetzt. Dabei waren sicherlich die während des Betriebs am Atari gesammelten Erfahrungen sehr nützlich, denn Datamat-Amiga ist nicht nur eine reine Umsetzung, sondern eine um einige Funktionen erweiterte Version.

Wie es sich für eine Neuentwicklung gehört, ist Datamat auf allen Amiga-Modellen mit Kickstart 1.2 lauffähig. Um ihn zu starten, klickt man entweder das Icon des Programmes an oder ruft es vom CLI aus auf. Nach dem Laden fällt auf, daß die Umsetzung vom Atari gut gelungen ist. Das voll menügesteuerte Programm nutzt mit der Verwendung von Menüs, Schaltsymbolen und Fenstern die vielfältigen Möglichkeiten der grafischen Benutzeroberfläche des Amiga. Zusätzlich lassen sich fast alle Funktionen auch über die Tastatur anwählen. Eine freie Belegbarkeit der Funktionstasten ermöglicht den Einsatz von Makrobefehlen.

Nach dem Start fordert das Programm zur Eingabe eines Dateinamens auf. Man kann dann entweder den Namen einer bereits existierenden Datei oder zu erstellenden Datei angeben. Ist der Dateiname unbekannt, so muß durch eine weitere Eingabe bestätigt werden, ob die Datei auch wirklich erstellt werden soll.

Bei der Definition einer Datenstruktur kann man pro Datei unbeschränkt viele Datenfelder einrichten. Jedes dieser Felder kann bis zu 32 000 Zeichen groß sein. Der Anwender hat die Auswahl zwischen sechs Datentypen: Textfeld, Zahlfeld, Datumsfeld, Zeitfeld, Auswahlfeld und IFF-Feld (Bild 1). Bei den Zahl-, Datums- und Zeitfeldern kann man beliebige Formate definieren, bei den Auswahlfeldern lassen sich die Worte, die für das jeweilige Feld erlaubt sind, vom Anwender bestimmen.

Der Datentyp IFF-Feld soll es ermöglichen, auch Grafiken zu verwalten. Laut Handbuch kann mit einer bestimmten Funktion der Inhalt solcher Dateien (Files) angeblich auch auf dem Bildschirm anzeigt werden. Die vorliegende Programmversion war nicht bereit, die Bilder anzuzeigen; diese Funktion scheint noch nicht implementiert zu sein.

Im Anschluß an die Definition der Felder lassen sich Paßworte vergeben, um die Daten vor unberechtigtem Zugriff zu schützen. Dabei kann man ein Paßwort für den Operator und eines für den Benutzer definieren. Der Operator hat alle Zugriffsrechte auf eine Datei, während der Benutzer die Datenstruktur nicht mehr ändern darf.



Bild 1. Festlegen der Dateistruktur bei Datamat

Nun sieht man auf dem Bildschirm die von Datamat erzeugte Standardmaske. Da diese Maske jedoch in den wenigsten Fällen den Ansprüchen des Benutzers genügen wird, kann man mit dem Maskeneditor die Maske auf sehr komfortable Weise verändern. Mit der Maus lassen sich alle Felder beliebig auf dem Bildschirm bewegen und in der Größe verändern. Der Anwender kann die verschiedensten Zeichensätze und Schriftarten in den Feldern benutzen sowie Farben und verschiedene Füllmuster für die Felder frei wählen (Bild 2). In der Maske kann gezeichnet werden. Dazu stehen Kreise, Ellipsen und Quadrate zur Verfügung. Natürlich lassen sich auch Texte in der Maske plazieren. Die Anzahl der maximal 5000 x 5000 Pixel großen Masken pro Datei scheint nur durch den Speicherplatz begrenzt zu sein. An der Größe können Sie erkennen, daß sich eine solche Maske über mehrere Bildschirmseiten erstrecken darf. Die einzige Schwachstelle des ansonsten hervorragenden Bildschirmmaskeneditors ist eine fehlende Funktion, mit dessen Hilfe sich für die Datenfelder bestimmte Formate, wie zentriert oder rechtsbündig, auswählen lassen.

Neben der Bildschirmmaske kann der Anwender auch noch Formularmasken und Listenmasken für die Ausgabe auf dem Drucker definieren. Für die Formularmaske steht eine ganze Seite zur Verfügung. Hier jedoch besteht nicht mehr die Möglichkeit, verschiedene Zeichensätze zu benutzen. Sofern der Drucker es unterstützt, kann aber unterstrichener, kursiv oder fett gedruckter Text ausgegeben werden.

Bei der Definition der Listenmaske steht nur eine Zeile für die Daten zur Verfügung. Dies ist für eine Liste auch vollkommen ausreichend.

Durch die Wahl des Menüpunktes »Dateneingabe« gelangt man in die vorher definierte Bildschirmmaske und kann mit der Dateneingabe beginnen. Neben der Bildschirmmaske befindet sich eine Leiste mit Symbolen, die die Bedienung bei der Eingabe vereinfacht. Der Anwender braucht um Daten zu speichern, zu löschen oder zu suchen nicht erst im Menü den Menüpunkt anwählen, sondern kann direkt durch Anklicken der betreffenden Symbole die Funktion auslösen.

Die Dateneingabe ist denkbar einfach: Nach einem Mausklick innerhalb des ausgewählten Datenfeldes erscheint der Cursor zur Eingabe. Nach der Eingabe eines Datenfeldes genügen die Return- oder Pfeiltasten für den Sprung in das nächste Datenfeld. Hat man einen Datensatz eingegeben, so löst ein Mausklick auf dem Ausrufezeichen der Symbolleiste, der Menüpunkt »Datensatz anhängen« oder die Tastenkombination <Amiga-A> die Speicherung des Datensatzes aus. Danach werden die Felder gelöscht und der nächste Datensatz kann eingegeben werden. Bei der Eingabe von Text in Datenfelder, die länger als eine Bildschirmzeile sind, wird am Zeilenende ein automatisches Wordwrap durchgeführt.

Ebenso einfach wie die Eingabe sollte laut Handbuch das Ändern von Datensätzen sein. Man müsse einfach im zu ändernden Datensatz den Cursor auf die zu ändernde Stelle setzen, die Änderung durchführen und abspeichern — schreiben die Programmautoren. Tatsächlich jedoch gelang es im Test nicht, auch nur einen einzigen Datensatz zu ändern, denn Datamat speicherte die Änderungen einfach nicht ab.

Um Daten zu suchen, schaltet der Anwender in den Suchmodus und kann dann in einer

leeren Maske die Suchkriterien bestimmen. Die Suchmöglichkeiten sind recht komfortabel. Die Suche nach teilweiser Übereinstimmung läßt sich mit dem »?« als einen Platzhalter für ein beliebiges Zeichen und mit »*« als Platzhalter für beliebig viele Zeichen durchführen. Weiterhin lassen sich bei den Zahlfeldern Intervalle bestimmen und bei Textfeldern kann nach dem Vorkommen des Textes an beliebiger Stelle im Datenfeld oder im gesamten Datensatz gesucht werden.

Befinden sich im Suchbegriff mehrere Texte durch ein »I« getrennt, so interpretiert Datamat jeden dieser Texte als einzelnen Suchbegriff. Auf diese Weise lassen sich mit dem Suchbegriff »MayerIMeierI MeyerIMaier« gleichzeitig alle Variationen dieses Namens ausfiltern. Man bezeichnet diese Arbeitsweise auch als Oder-Verknüpfung, da das Programm Mayer oder Meier oder. . . findet. Natürlich können diese Namen auch mit »M??er« gesucht werden; allerdings findet das Programm dann auch »Maler« und ähnliche. Die Kombination aller Suchkriterien ist möglich und erlaubt so sehr genaues Suchen ohne viel Aufwand.

Um die Suche zu beschleunigen, kann man auch einzelne Datenfelder als Indexfelder definieren. Dann wird die Datei in auf- oder absteigender Reihenfolge nach dem Indexfeld sortiert und die Feldinhalte der Indexfelder werden im Speicher gehalten. Sucht man nun nach einem Indexfeld, so wird nicht mehr die ganze Datei auf der Diskette durchsucht, sondern Datamat sucht den Feldinhalt in den im Speicher gehaltenen Indexfeldern. Dies beschleunigt natürlich die Suche enorm. Die beim Datamat fehlende Funktion, dieses Indexfeld nach Änderungen auf der Diskette zu speichern, ist zweifellos ein Nachteil. Wenn man am nächsten Tag wieder mit der Datei arbeiten will, muß man zunächst den Index neu erzeugen. Dadurch ist es nicht möglich, für bestimmte Dateien eine feste Ordnung vorzugeben, wie etwa für Adreßdateien, die nach dem Nachnamen sortiert sein sollen.

Ein weiteres Manko ist die fehlende Möglichkeit der zweistufigen Indizierung. Eine Datei läßt sich also nicht zuerst nach den Nachnamen und dann nach den Vornamen ordnen. Dies ist zwar über Selektion möglich, doch werden

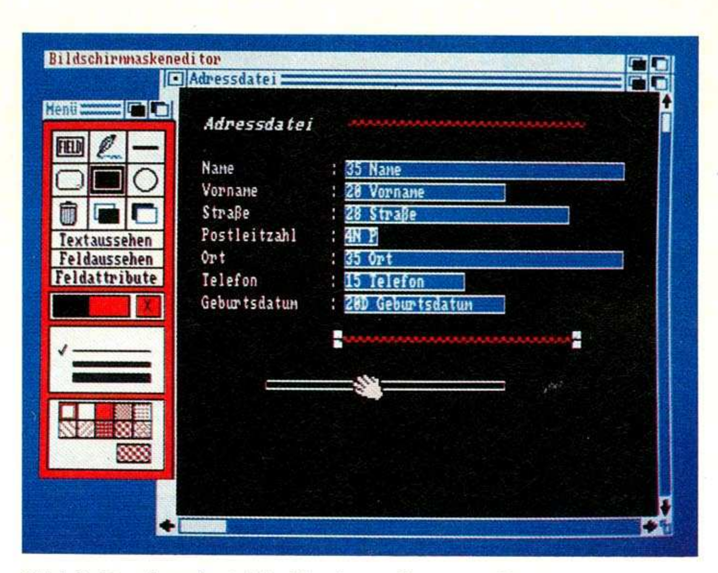


Bild 2. Der komfortable Maskeneditor von Datamat

Neueingaben natürlich nicht sofort richtig eingeordnet, wie es bei der Indizierung der Fall wäre.

Bei der Arbeit mit großen Datenbeständen ist für bestimmte Aufgaben manchmal nur eine Teilmenge dieser Daten erforderlich. So können zum Beispiel für eine regionale Werbekampagne nur diejenigen Kunden der Adreßdatei von Interesse sein, die im Postleitzahl-Bereich 3000 bis 4000 wohnen. Eine Reduzierung des Datenbestandes auf eine Teilmenge bezeichnet man als Selektieren. Das gewählte Selektierkriterium wäre in diesem Fall der Postleitzahlen-Bereich. Nach einer Selektion kann man mit diesen Daten in gewohnter Weise arbeiten. Hebt man die Selektierung auf, hat man wieder Zugriff auf den gesamten Datenbestand. Funktionen zum Speichern der Selektierkriterien und der selektierten Datensätze erleichtern die Arbeit mit diesen Datengruppen.

Für den Austausch von Datensätzen innerhalb mehrerer Dateien nutzt Datamat ebenfalls die Selektierfunktion. Selektierte Daten lassen sich sequentiell abspeichern und in eine andere Datei einlesen. Auf ähnliche Weise ist auch der Datenaustausch zwischen Datamat und anderen Programmen zu realisieren. Dabei werden die Daten als IFF-File im ASCII-Format mit frei wählbaren Trennzeichen ausgegeben. Der Erzeugung von Serienbriefen über Mailmerge-Dateien steht so nichts mehr im Wege.

Wie bereits erwähnt, lassen sich die Zugriffsrechte für die Dateien durch die Vergabe von Paßwörtern einschränken. Nutzt der Anwender die Funktion nicht, so werden die Daten unverschlüsselt gespeichert. Verwendet er das Paßwort, so werden die Daten verschlüsselt gespeichert und sind damit zumindest vor Zugriff durch den Befehl TYPE sicher.

Um die eingegebenen Daten mittels einer definierten Formularmaske auf dem Drucker auszugeben, gibt es die Funktion »Formular ausgeben«. Die Kommunikation mit dem Drucker funktioniert ohne Probleme. Datamat unterstützt alle Druckertreiber und die Ausdrucke stimmen genau mit den Masken überein. Im Testbetrieb passierte es jedoch häufig, daß Datamat, obwohl schon alle Daten ausgegeben waren, immer weiter leere Seiten druckte. Nach Wahl des Befehls »Abbrechen« stürzte das Programm dann ab. Scheinbar wurden hier Programmteile unsauber programmiert, denn auch an anderer Stelle war dieses unsaubere Programmende zu beobachten.

Diese Schwächen mindern leider den Wert des Programmes beträchtlich, denn es können leicht größere Datenmengen verlorengehen, wenn der Anwender mit der RAM-Disk arbeitet und das Betriebssystem anfängt zu »meditieren«.

Datamat wird mit einem 250-seitigen Handbuch mit Kartoneinband geliefert. Das Handbuch richtet sich vom Aufbau her eindeutig an die Einsteiger in die Dateiverwaltung. Um dem Anwender die Einsicht in die mit dem Computer gelieferte Dokumentation zu ersparen, beginnen die Autoren des Handbuches zunächst mit der Bedienung des Amiga. Im Anschluß daran wird das Prinzip und wesentliche Begriffe der elektronischen Dateiverwaltung erläutert und dann folgt die schrittweise Einführung in das Programm. Der fortgeschrittene Leser wird die eine oder andere Einzelheit allerdings vermissen, denn einige interessante Funktionen, wie etwa das Verknüpfen mehrerer Dateien, werden nur sehr kurz behandelt. Der Anhang des Handbuches enthält genaue Informationen über die verschiedenen, von Datamat erzeugten Dateien und das Format der Daten auf Diskette. In dem knapp gehaltenen Index sucht man so manchen Begriff allerdings vergebens.

Die Programmidee, die gelungene grafische Benutzerführung und die vielfältigen Möglichkeiten, die das Programm bietet, zeigen doch, daß hier erfahrene Entwickler am Werk waren. Dieser hervorragende Gesamteindruck wird aber stark getrübt durch Schwächen in der Umsetzung, wie etwa der Absturzhäufigkeit und den nicht implementierten Funktionen zur Datenänderung und zum Anzeigen der IFF-Grafiken. All diese Macken hätten sicherlich in einer längeren Testphase behoben werden können. Der Anwender kann auf eine überarbeitete Version gespannt sein.

(Michael Bass/pa)

AMIGA-WERTUNG

Software: Datamat

7,9 von 12	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■	■	
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■			

Fazit: Von der Konzeption her gut für die Verwaltung umfangreicher Datenbestände geeignet; durch die fehlende Implementation beziehungsweise Betriebssicherheit mancher Funktionen aber nicht voll einsatzfähig.

Positiv: Auswahlfelder; komfortable Suchfunktionen; Maskeneditor; auf Einsteiger abgestimmte Dokumentation; einfache Bedienung bei der Eingabe.

Negativ: Index nicht speicherbar; keine mehrstufige Indizierung; nicht absturzsicher.

DATEN

Produkt: Datamat

Preis: 99 Mark

Hersteller: Data Becker GmbH

Anbieter: Data Becker GmbH; Merowinger Str. 30; 4000 Düsseldorf; Tel. 0211/310 01 30.

Sie haben einen Amiga, Atari ST oder QL? Wir haben die Zeitschrift dazu!

Markt&Technik
9/87 68000er
DAS MAGAZIN DER NEUEN COMPUTER-GENERATION
SEPTEMBER 1987
DM 7,–

Grafik
- Test: die besten Programme
- Einführung: meisterhaft malen
- Programmierung: Blitter

Amiga
- Zum Mega-Amiga aufrüsten
- Tolles DFÜ-Duo

Atari ST
Tempo für Basic: Neuer Compiler

QL
Hardware klar und verständlich

Viele anspruchsvolle Listings zum Abtippen

Amiga, Atari ST, QL — die Spitzentechnologie im »68000er«

»68000er«, das Magazin der neuen Computer-Generation, bringt ausführlich Berichte, Tests und Beschreibungen von

- Hard-, Software und Peripherie
- Programmiersprachen: Kurse in Basic, C, Modula 2, Assembler und Pascal
- Homeentertainment: aktuelle Spiele, neue Grafikprogramme
- Tips & Tricks und Bauanleitungen für Hardware-Erweiterungen

Machen Sie sich die Technologie des Amiga, Atari St und QL sofort verfügbar – nutzen Sie die Chance, mit einem kostenlosen Probeexemplar das »68000er«-Magazin kennenzulernen.



KENNENLERN-ANGEBOT

MIT EINEM KOSTENLOSEN »68000er«-PROBEEXEMPLAR

JA, ich möchte »68000er«, das Magazin der neuen Computer-Generation, kennenlernen. Senden Sie mir bitte die aktuelleste Ausgabe kostenlos als Probeexemplar. Wenn mir »68000er« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte es dann regelmäßig frei Haus per Post. Außerdem nutze ich den Abonnement-Preisvorteil von 8% und bezahle pro Jahr nur 77,– DM statt 84,– im Einzelverkauf.

Name, Vorname

Straße

PLZ, Ort

Datum, 1. Unterschrift

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

Datum, 2. Unterschrift.

Gutschein ausfüllen und absenden an: Markt & Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar bei München

3D-Zeichnen mit Master-CAD

AMIGA test

Mit »Master-CAD« stellt erstmals eine deutsche Firma ein CAD-Programm für den Amiga vor. Im Gegensatz zu anderen neuen 3D-Programmen steht bei Master-CAD allerdings nicht die Animation, sondern die komfortable Erstellung von Objekten im Mittelpunkt.

Das Anwendungsgebiet von Master-CAD liegt bei der Darstellung von beliebigen dreidimensionalen Körpern als Drahtgittermodell oder schattierte Grafik. Um mit »Master-CAD« ein Objekt dreidimensional darstellen zu können, muß dem Programm natürlich zuerst einmal die Form des Objektes mitgeteilt werden. Das geschieht mit Hilfe des Objekt-Editierprogrammes »Builder« (Bild). Er ist eines der wenigen Programme, das die bei deutschen Amiga-Modellen vorhandenen zusätzlichen Bildschirmzeilen der PAL-Auflösung ausnutzt. Die Eingabe wird größtenteils mit Hilfe der Maus durchgeführt, was das Eintippen langer Zahlenreihen mit Koordinaten erspart.

Dimensionieren mit der Maus

Damit beim Arbeiten an einem dreidimensionalen Objekt die Übersicht nicht verlorengeht, wird es in einem Fenster ständig von vorne (Bild links) und in einem anderen Fenster ständig von der Seite her (Bild rechts) gezeigt. Die beiden Fenster, in denen das Objekt zu sehen ist, dienen außerdem zur Festlegung der Zeichenposition innerhalb des dreidimensionalen Raumes. Dazu wird zuerst mit der Maus im rechten Fenster die gewünschte Tiefe (Z-Koordinate) im Raum angewählt, wobei das Programm diese Koordinate ständig als Zahlenwert anzeigt. Dann klickt man im linken Fenster mit dem Mauszeiger auf den gewünschten Punkt innerhalb der vorher gewählten Zeichenebene, was durch eine Anzeige von X- und Y-Koordinate erleichtert wird. Dieses Eingabeprinzip — zuerst Tiefe festlegen, dann Punkt innerhalb der festgelegten Ebene setzen — wird bei allen Zeichenfunktionen (Punkt, Linie, Rechteck, Polygon und Freihandzeichnen) verwendet. Zusätzlich zu den normalen Zeichenfunktionen ist noch eine Rotationsroutine vorhanden, die Rotationskörper automatisch erstellt.

Der gezeichnete Körper kann nun durch einfaches Anklicken der Plus- und Minus-Zeichen, bei »MOVE«, »ROT« und »SIZE« beliebig im Raum verschoben, gedreht, vergrößert und verkleinert werden.

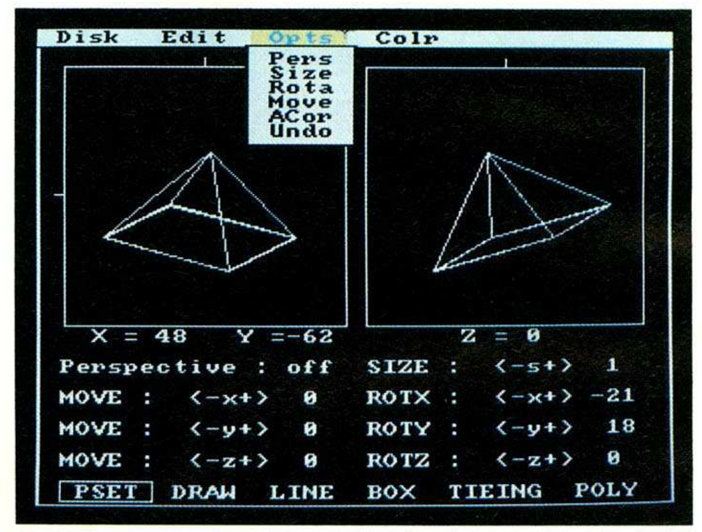


Bild. Der Hauptteil von Master-CAD ist der »Builder«

Allein mit der Maus ist es allerdings nicht möglich, während des Zeichnens die Zeichentiefe zu verändern, um zum Beispiel eine Linie quer durch den dreidimensionalen Raum zu legen. Damit man die Zeichentiefe auch während des Zeichnens von Linien und beim freihändigen Zeichnen ändern kann, wird in der Anleitung die »Pfeil größer« und »Pfeil kleiner«-Taste angeführt. Das Drücken der mit diesen beiden Zeichen beschrifteten Taste hat aber keinerlei Wirkung — es bleibt dem Benutzer des Programms überlassen, herauszufinden, daß die Komma- und Punkt-Taste betätigt werden muß.

Gut gelungen ist die UNDO-Funktion des Programms, die es nicht nur erlaubt, den jeweils letzten, sondern beliebig viele Zeichenschritte zurückzunehmen. Da der Builder aber nicht ganz absturzsicher ist, sollte man das Objekt gelegentlich speichern.

Ist das Objekt schließlich konstruiert, so wird es auf Diskette gespeichert und der »Depicter« geladen. Dieses Programm dient dazu, die mit dem »Builder« erstellten Objekte als Draht- oder schattierte Grafik im Low-Resolution-Modus auf den Bildschirm zu bringen, wobei die zusätzlichen Bildschirmzeilen aber ungenutzt bleiben. Dabei werden die nicht sichtbaren Linien automatisch unterdrückt (Hidden-Line-Modus). Ein gerade geladenes Objekt wird immer mit seiner bereits im Builder eingegebenen Rotation auf den Bildschirm gebracht; die Rotationswinkel können aber auch als Zahlenwerte neu eingegeben werden.

Damit die vom »Depicter« erstellten Bilder mit einem Malprogramm weiterverarbeitet werden können, läßt sich der Bildschirminhalt des Depicters im IFF-Format speichern, das von allen Malprogrammen verstanden wird. Beim Speichern im IFF-Format kommt es allerdings manchmal zu einem Programmabsturz.

Alle Funktionen des Depicters und viele Funktionen wie Laden, Speichern, Undo und Farbwahl des Builders werden über die Amiga-bekannten Menüs gesteuert. Offensichtlich wollten die Autoren von »Master-CAD« bei ihrem Programm aber wohl etwas zuviel Speicherplatz sparen, denn sämtliche Menüeinträge sind Abkürzungen aus vier Buchstaben. Das macht die Einarbeitungszeit in »Master-CAD« länger als nötig, denn wer kann sich schon auf Anhieb merken, was »FCol«, »ResC«, »ColM« oder »ACor« bedeuten?

Das Arbeiten mit Disketten ist in Master-CAD unpraktisch gelöst. Es wird nicht wie bei fast allen Amiga-Programmen über die Maus gesteuert, sondern erfordert die Eingabe der Namen der Verzeichnis- und Objektnamen über die Tastatur.

Master-CAD ist eines der wenigen Programme mit deutschem Handbuch. Leider ist das fünfzigseitige Heft aber teilweise nicht sehr logisch aufgebaut (die Feinheiten der UNDO-Funktion werden zum Beispiel schon vor der Behandlung dieser Funktion selbst erklärt).

Insgesamt ist Master-CAD ein Programm, mit dem sich durchaus professionell wirkende dreidimensionale Bilder erstellen lassen.

(Andreas Lietz/jk)

AMIGA-WERTUNG

Software: Master-CAD

6,6 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■				
Dokumentation	■	■	■	■		
Bedienung	■	■	■			
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Master-CAD bietet für Einsteiger in den Bereich CAD eine Fülle von Funktionen, dreidimensionale Körper zu erstellen und anzuzeigen. Für den relativ hohen Preis könnte man aber mehr Bedienungskomfort erwarten.

Positiv: Deutsche Anleitung; Objekte mit Maus erstellbar; Relativ schnelle Grafikberechnung; kein Kopierschutz.

Negativ: Menüsteuerung mit Abkürzungen ist schwer erlernbar; Arbeit mit Disketten umständlich; nicht ganz absturzsicher; Schattierungen nur in Grautönen.

DATEN

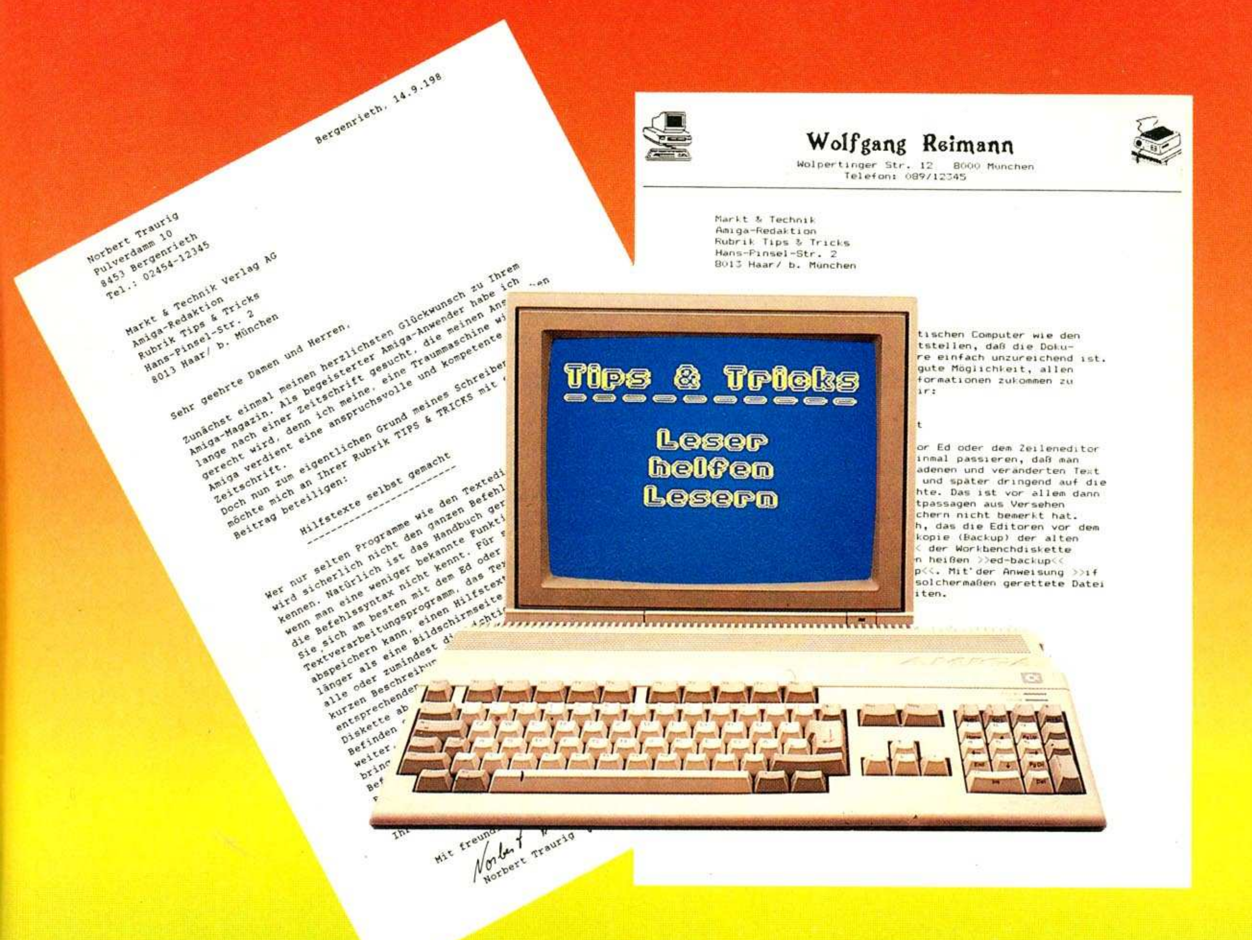
Produkt: Master-CAD

Preis: 238 Mark

Hersteller/Anbieter: Atlantis, Dunantstr. 53, 5030 Hürth, Tel. 02233/41081

Gesucht:

Tips & Tricks zum Amiga



Sie sind ein begeisterter Amiga-Fan? Sie kennen Ihren Computer inzwischen so gut, daß die Begriffe »CLI«, »Amiga-DOS« oder »Workbench« für Sie keine Fremdwörter mehr darstellen? Dann beteiligen Sie sich doch an der Tips & Tricks-Rubrik »Ihres« neuen Magazins!

Die Tips & Tricks-Ecke im Amiga-Magazin ist eine reine »von-Lesern-für-Leser«-Rubrik. Das bedeutet, daß Sie durch Ihre Beiträge und durch Ihre guten Ideen anderen Lesern helfen können, mit dem neuen Computer besser zurechtzukommen. Sicherlich erinnern Sie sich noch an die Zeit, wo Sie begierig jede Zeile Information zum Amiga mit Heißhunger verschlangen. Gerade als Einsteiger ist man auf die Erfahrung und das Wissen von »Profis« angewiesen.

Sie haben eine neue Idee, wie man mit einem kleinen Amiga-DOS-Programm die Arbeit mit dem Computer vereinfachen kann? Einschicken!

Sie haben ein raffiniertes, kurzes Basic-, C- oder Assembler-Programm ausgeklügelt? Einschicken!

Sie haben einen Trick herausgefunden, wie man mit der einen oder anderen Software komfortabler arbeiten kann? Einschicken!

Sie haben Ihren Amiga durch eine kleine Bastelei aufgewertet? Einschicken! Jede gute Idee wird in »Ihrer« Rubrik veröffentlicht. Natürlich bekommen Sie dann von uns ein entsprechendes Honorar. Wichtig ist nur, daß Sie in einer kurzen Beschreibung den Grundgedanken Ihrer Idee zu Papier bringen. Wenn Sie ein Programm einschicken, sollten Sie nicht vergessen, eine Diskette mitzuschicken. Und sollte einmal ein Beitrag, aus welchen Gründen auch immer, nicht zur Veröffentlichung geeignet sein, bekommen Sie Ihre gesamten Unterlagen natürlich wieder zurückgesandt.

Schreiben Sie an:

Markt & Technik Verlag AG
Redaktion Amiga-Magazin,
Aktion Tips & Tricks
Hans-Pinsel-Str. 2
8013 Haar bei München

Eines sollten Sie nicht vergessen: Sie müssen kein Vollprofi sein, um bei dieser Aktion mitzumachen. Auch Tricks, die für Sie schon längst zur Routine geworden sind, könnten anderen Lesern eine wichtige Hilfe sein. Vor allem kurze und leicht durchschaubare Listings sind begehrt. Wir wünschen Ihnen viel Spaß beim Programmieren und Knobeln! (ub)

Wohin des Weges, Abenteurer?

AMIGA test

Tief in die Gruft eines längst vergessenen Druiden oder ins verwunderliche Gnomenland führen zwei neue Grafikabenteuer bekannter Softwarefirmen. Ohne Magie kommt man dabei nicht aus.

Langsam aber unaufhaltsam werden auch einige der renommiertesten Softwarehäuser und Spieleproduzenten auf den Amiga aufmerksam. Mit der tollen Grafik und dem schnellen Prozessor kann man schon sehr gute Spiele produzieren, die dann natürlich auch für entsprechenden Umsatz sorgen. Diesmal haben sich eine Firma aus England (Level 9) und eine aus Amerika, nämlich Aegis, entschlossen, ein Text/Grafikadventure für die Abenteuerinteressierten unter den Spielefreunden zu schreiben. Level 9 dürfte einigen Bekanntheitsgrad durch pfiffige Adventure auf dem C 64 haben und hat auch schon zwei einschlägige Spiele auf dem Amiga veröffentlicht: »Jewels of Darkness« und »Knight Orc«. Das dritte Abenteuer ist das neue »Gnome Ranger«, das sich von der Gestaltung her stark an Knight Orc orientiert und hier zum Test vorliegt. Das andere Spiel, um das es geht, ist Aegis' bisher einziges Adventure »Arazok's Tomb«.

In letzterem übernehmen Sie die Rolle eines Sensationsreporters, der das Geheimnis eines alten schottischen Druidengrabes erforschen soll. An dieses »Cear Arazok« genannte Grab traut sich keiner der Bewohner des nahegelegenen Ortes New Galloway. Vor langer Zeit soll die Priestersekte unter ihrem Herren Arazok dort ziemlich dunkle Machenschaften veranstaltet haben. Bis sich einige vernünftige Priester gegen den Meister gewandt und ihn mit einem Trick in die Hölle verbannt haben.

Ab in die Gruft

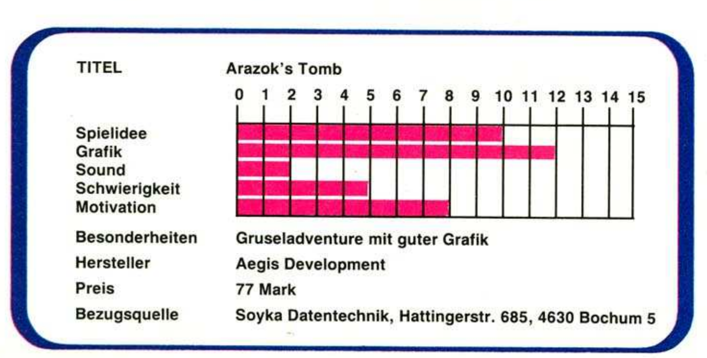
Obwohl der Platz verrufen ist, hat sich vor kurzer Zeit eine Expedition aufgemacht, die Gruft zu erforschen und ist natürlich nicht zurückgekehrt. Die letzte Meldung des Forscherteams, ein Telegramm von Daphne, der Tochter des Expeditionsleiters, hat die Öffentlichkeit alarmiert: Was geht tatsächlich in den unbekannten Tiefen vor? Mit dieser Ausgangssituation startet das Adventure und als erstes muß man herausfinden, wie man nun überhaupt in die Gruft hineinkommt. Dies scheint auf den ersten Blick gar nicht so schwierig zu sein. Muß man doch dazu nur in die steinerne Statue des bösen Arazok, die den Eingang versperrt, ein rotes Juwel einsetzen. Doch daß hier die ersten Schwierigkeiten auftauchen, liegt nicht etwa am Adventure selbst, sondern am viel zu einfachen Parser. Dieser Programmteil, der die Texteingaben des Spielers auswertet, ist den Aegis-Programmierern etwas sehr dürftig geraten. Er versteht viel zu wenig Worte und entsprechende Synonyme und beherrscht auch nur einfache Satzkonstruktionen. So wird jede Aktion im Spiel zu einem mühsamen Suchen nach genau dem richtigen Wort oder einer vorgegebenen Satzstellung, um das Gewünschte zu erreichen. Das läßt leider keinen besonders guten Spielfluß aufkommen und macht das Adventure schwerer als es eigentlich ist. Viele Probleme sind nämlich auf ganz offensichtliche Weise zu lösen. Hieran erkennt man eindeutig, daß dies Aegis' erstes Adventure ist. Vielleicht wird bei weiteren Abenteuern aus diesem Softwarehaus auf solche Sachen mehr geachtet. Ein Adventure, bei dem man die wesentlichen Spielzüge durch die Eingabe von Texten vollzieht, steht und fällt nun einmal mit diesem Parser. Auch wenn das Debüt in dieser Hinsicht sehr schlecht ausgefallen ist, um so mehr überzeugt Arazok's Tomb durch seine gute Grafik und Benutzerführung. Alle Bilder der verschiedenen Räume sind vom farblichen und grafischen Gesichtspunkt hervorragend gezeichnet und vermitteln einen guten Eindruck des Ortes, an dem man sich gerade befindet. Einzelne Objekte, die herumliegen, werden beim Aufnehmen aus dem Bild entfernt und tauchen wieder auf, sobald man sie ablegt. Außerdem sind manche Räume, die sich verändern können oder in denen sich etwas Bewegtes befindet,



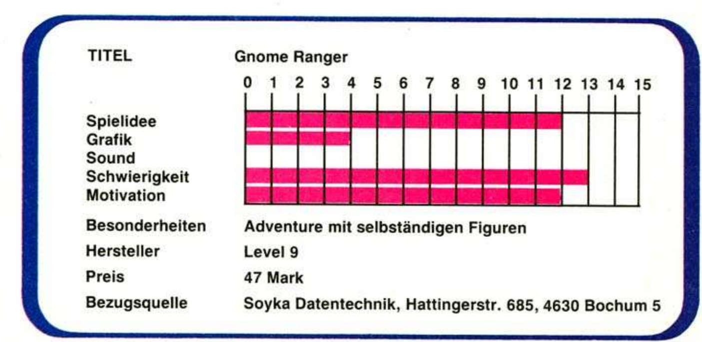
Bild 1. Die Schlange in »Arazok's Tomb« versperrt den Durchgang, hat aber Angst vor dem Zauberstab



Bild 2. Letzter Gegner ist der böse Zauberer selbst. Wenn man ihm begegnet, sollte man das Prisma dabeihaben.



TITEL	Arazok's Tomb
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Gruseladventure mit guter Grafik
Hersteller	Aegis Development
Preis	77 Mark
Bezugsquelle	Soyka Datentechnik, Hattingerstr. 685, 4630 Bochum 5



TITEL	Gnome Ranger
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Adventure mit selbständigen Figuren
Hersteller	Level 9
Preis	47 Mark
Bezugsquelle	Soyka Datentechnik, Hattingerstr. 685, 4630 Bochum 5

GEWINN: DM 2000,–



FÜR DAS BESTE PROGRAMM DES MONATS

2000 Mark ist uns Ihr Programm wert, wenn es als Programm des Monats im Amiga-Magazin veröffentlicht wird. — Haben Sie für Ihren Amiga ein Super-Programm geschrieben?

Dann gibt es nur eines: einschicken an die Redaktion des Amiga-Magazins. Wir wählen aus den besten Listings, die wir veröffentlichen, ein Programm des Monats aus, das mit 2000 Mark honoriert wird.

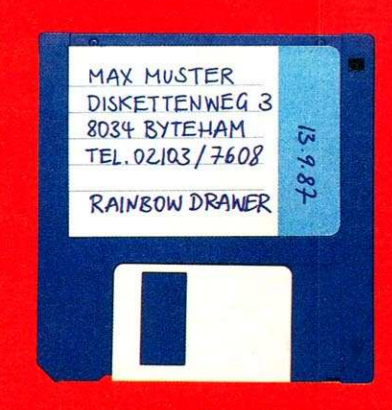
Für Sie also eine Mühe, die sich lohnt. Ob Sie nun ein Action-Spiel oder eine ernsthafte Anwendung auf Lager haben, gute Programme sind immer willkommen.

Haben Sie sinnvolle und mächtige Betriebssystem-Erweiterungen programmiert, die die Bedienung des Amiga vereinfachen, hier haben Sie die Chance, anderen Ihre Entwicklung zugute kommen zu lassen und auch noch etwas dabei zu verdienen.

Schicken Sie Ihr Programm an:

Markt & Technik Verlag AG
AMIGA-Redaktion: Programm des Monats
Hans-Pinsel-Str. 2, 8013 Haar bei München

Ihr Programm im Amiga-Magazin



Haben Sie bereits ein interessantes Programm für den Amiga geschrieben? Dann sollten Sie sich vielleicht überlegen, ob das von Ihnen entworfene Programm nicht für die Allgemeinheit der Amiga-Besitzer nützlich ist.

Nicht nur neue Spiele werden durch neue und leistungsfähige Computer wie den Amiga zu faszinierenden Meilensteinen in der Heimcomputer-Geschichte. Durch die steigende Speicherkapazität und die immer professionelleren Betriebssysteme können auch immer mehr höhere Programmiersprachen eingesetzt werden. Da die einzelnen Systeme heute immer komplexer aufgebaut sind, werden sogar Betriebssysteme in einer höheren Sprache erstellt. Auch hier ist der Amiga wieder ein sehr gutes Beispiel. Seine Seele, das Betriebssystem, wurde mit der modernen Sprache C programmiert.

Da für den Amiga in der Zwischenzeit ein bunt gemischtes Sammelsurium an verschiedensten Sprachen zur Verfügung steht, von Fortran bis Prolog, versorgen wir Sie natürlich mit brandheißen Informationen zu diesen Sprachen. Dabei wollen wir selbstverständlich nicht an unseren Lesern »vorbei programmieren«. Vielmehr haben Sie hier die Möglichkeit, Ihr Amiga-Magazin als Leser mitzugestalten. Wir suchen vor allem kurze Programme mit großer Wirkung, also Utilities oder wirkungsvolle Routinen, die jeder Leser und Programmierer sehr leicht für sich aus- und verwerten kann. Besonders nützlich ist auch eine genaue Dokumentierung des Quellcodes. Dadurch erschließt sich die von Ihnen verwendete Programmtechnik auch allen anderen Amiga-Anwendern. Die Programme werden leicht nachvollziehbar, und bestimmte Teile können so auf einfache Weise in eigene Programme eingebaut werden. So steigert sich ganz allmählich auch die Qualität der veröffentlichten Programme. Natürlich wird Ihr Beitrag bei Veröffentlichung entsprechend honoriert. Welche Sprache oder welchen Compiler Sie dabei verwenden, ist vollkommen egal. Ob Sie nun das bereits vorhandene Amiga-Basic benutzen oder sich mit Maschinensprache und C herumschlagen, wir werden Ihre Einsendungen bewerten und, wenn für gut befunden, in einer der folgenden Ausgaben veröffentlichen. Wie gesagt, sie können jede Sprache verwenden, die für den Amiga erhältlich ist.

Wie gehen Sie bei der Einsendung vor? Ganz einfach, schicken Sie uns Ihr Programm mit genauer Beschreibung und Diskette, sowie der Angabe der verwendeten Sprache an folgende Adresse:

Markt & Technik Verlag AG
Amiga-Redaktion
Stichwort:
Amiga-Programmierung
Hans-Pinsel-Str. 2

8013 Haar bei München



Bild 3. »Gnome Ranger« bietet digitalisierte Grafiken

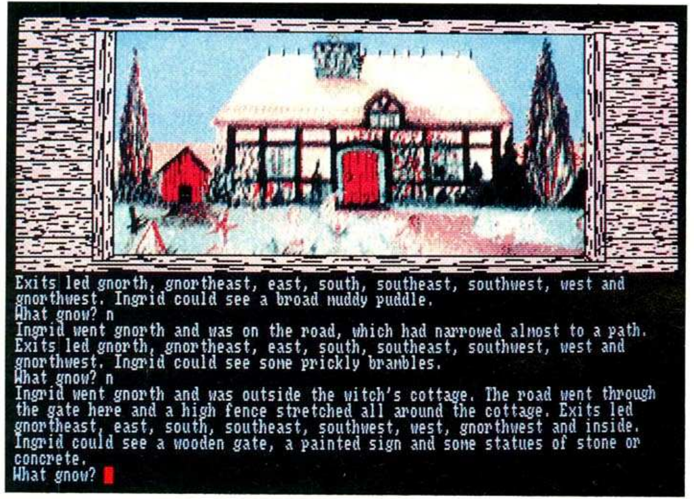


Bild 4. So lieblich sieht das Häuschen aus, obwohl eine gefährliche und häßliche Hexe darin wohnt

mit einer animierten Grafik versehen. Außer dem Text- und Grafikfenster ist auf dem Arazok-Bildschirm noch ein Feld mit einem kompaßähnlichen Richtungszeiger und einer Uhr angebracht (Bild 1). Durch einfaches Anklicken mit der Maus kann man in die gewünschte Richtung gehen.

Abenteuer für Einsteiger?

Gut gelöst ist auch, daß man den Kommentartext in seinem Fenster mit Scroll-Pfeilen beliebig auf und ab schieben kann (Bild 2). Zur komfortablen Bedienung trägt weiterhin die Verwaltung der gespeicherten Spiele sowie das automatische Laden des zuletzt gespeicherten Spielstandes bei.

Von der Schwierigkeit der gestellten Aufgabe, der Grafik und der guten Bedienung her gesehen, ist Arazok's Tomb eigentlich eher ein Adventure für Einsteiger. Wirklich empfehlen kann man es jedoch nur erfahrenen Abenteurern, die nicht schon nach dem ersten Frust mit dem Parser das Handtuch werfen und sprachliche Klippen elegant umschiffen können.

Ganz anders ist dagegen »Gnome Ranger« von Level 9 aufgebaut. Grundsätzlich ist dies ein Textadventure, dem für einige, aber nicht alle Räume ein digitalisiertes Bild mitgegeben wurde. Über die Qualität dieser Bilder läßt sich streiten. Tatsache ist aber, daß manche Bilder sehr verwaschen aussehen und einzelne Details, wie etwa Gegenstände zum Aufnehmen, überhaupt nicht zu sehen sind. Die Bilder sollen nur einen groben Einblick in die Stimmung einer jeweiligen Szene geben. Mit dem Mauszeiger kann man sie deswegen auch völlig nach oben aus dem Bildschirm hinausschieben und das Abenteuer nur mit Text weiterspielen. Außer einer knappen Anleitung wird dem Spiel ein Gnom-Tagebuch beigelegt, welches in das Spiel einführt und einige Hintergrundinformationen gibt. Man muß allerdings schon etwas besser Englisch sprechen, um damit klarzukommen. Wie das Tagebuch sind auch die Kommentartexte entsprechend ausführlich, aber mit vielen Spezialausdrücken gespickt. Für das Spiel muß man sich in eine etwas ungewohnte Rolle hineinversetzen. Die Hauptperson ist Ingrid, ein weiblicher Gnom, der durch einen mißratenen Zauberspruch in eine fremde Gegend teleportiert wurde. In diesem wilden Land klarzukommen, ist gar nicht so einfach (Bild 3). Gibt es doch gefährliche Nebel, in denen man sich hoffnungslos verirrt oder böse Hexen, die nichts Besseres zu tun haben, als einen sofort in eine steinerne Statue zu verwandeln (Bild 4). Niemand ist allerdings völlig verloren, wenn er sich auf Freunde verlassen kann. Ein Weg aus den Sumpfnebeln findet sich beispielsweise indem man einem am Himmel vorbeifliegenden Adler folgt.

Starker Parser

Das Adventure hat viele völlig eigenständig agierende Figuren, mit denen auch eine Unterhaltung möglich ist. Sie können sogar dazu aufgefordert werden, bestimmte Aufgaben zu übernehmen, für die ein Gnom nicht geschaffen ist, ohne deren Lösung man aber das Spiel nicht beenden kann. Um dies überhaupt erst zu ermöglichen, braucht das Adventure natürlich einen guten Parser; und es hat einen wirklich hervorragenden erhalten. Viele Worte und lange Sätze werden sehr schnell verarbeitet. Hier zeigt sich die durch viele Programme und Umsetzungen gewonnene Erfahrung des Level-9-Teams. So kommt mehr Spielspaß auf, auch wenn die gestellten Probleme viel schwieriger sind als bei Arazok's Tomb.

Sehr hilfreich beim Spielen ist die »RAM Save«-Funktion, mit der sich jederzeit der Spielstand in Sekundenschnelle im RAM speichern und laden läßt. So können kurzzeitig Fehlzüge zurückgenommen werden. Warum man sich an Gnome Ranger als Einsteiger nicht unbedingt heranwagen sollte, liegt nur in der Schwierigkeit des Abenteuers begründet. Für Adventure-Freaks, die sich an der verwaschenen Grafik nicht stören, ist Gnome Ranger jedoch das richtige. (jk)

Inserenten

AMIGA Markt&Technik

PROGRAMM-SERVICE

Super-Kopierprogramm mit viel Komfort

DCopy: Unser Programm des Monats, ein Kopierprogramm, das alles bietet, was man sich nur wünschen kann. Einige Fähigkeiten: Bis zu vier Laufwerke werden verwendet, Mehrfachkopien, abschaltbares Verify und vieles mehr. **SpeedHc:** Eine sehr schnelle Hardcopyroutine für Schwarzweißausdrucke mit höchster Qualität. Leicht an andere Drucker anzupassen. **Sternenhimmel:** Ein unentbehrliches Werkzeug für alle Himmelsbeobachter. Das Programm zeigt alle Sterne und Planeten von jedem beliebigen Punkt der nördlichen Hemisphäre. **Checkie42:** Der Checksummer für alle Programmiersprachen von Assembler über Basic bis zu C. Ab dieser Ausgabe finden Sie bei jedem Listing die Prüfziffern. **Joy:** Ein sehr kurzes und schnelles C-Programm zur Abfrage des Joysticks. Es ist leicht in eigene Programme einzubinden. **Amiga-Shell:** Ein C-Programm, das Komfort ins CLI bringt. Editieren der Befehlszeile, Funktionstastenbelegung und Aliasnamen sind nur einige Fähigkeiten dieses fantastischen Programms.

Eine Diskette für Amiga

Bestell-Nr. 48705 **DM 29,90 *** (sFr 24,90*/öS 299,-*)
* Unverbindliche Preisempfehlung

IFF-Grafik-Show für Anspruchsvolle

IFF-Diashow: Als Resultat des in Ausgabe 6-7/87 gestarteten Malwettbewerbs präsentieren wir Ihnen zwei randvoll bespielte Disketten, die mit den besten Bildern des Wettbewerbs (siehe Amiga-Magazin 10/87, Seite 129) bespielt sind. Die Bilder können entweder mit dem enthaltenen Diashow-Programm angesehen oder mit jedem gängigen IFF-Malprogramm geladen werden. Lassen Sie sich überraschen, welche hochwertigen und vor allem sehenswerten Grafiken unsere begabten Leser geschaffen haben.

Zwei Disketten für den Amiga

Bestell-Nr. 49705 **DM 29,90 *** (sFr 24,90*/öS 299,-*)
* Unverbindliche Preisempfehlung

Die Disketten aus dem Programm-Service erhalten Sie direkt bei Markt&Technik oder im ausgewählten Fachhandel



Weitere Angebote auf der Rückseite!

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

Postscheckkonto Nr. des Absenders

Empfängerabschnitt

DM | Pf

für Postscheckkonto Nr.
14 199-803

Lieferanschrift und Absender der Zahlkarte

PLZ Ort

Verwendungszweck
M&T Buchverlag
Programm-Service

Meine Kunden-Nr.:

DM | Pf | für Postscheckkonto Nr. 14 199-803

Absender der Zahlkarte

PSchA — Postscheckkonto Nr. des Absenders — Postscheckteilnehmer

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

Zahlkarte/Postüberweisung

DM | Pf | (DM-Betrag in Buchstaben wiederholen)

für Markt&Technik Verlag Aktiengesellschaft

in 8013 Haar

Postscheckkonto Nr.
14 199-803

Postscheckamt
München

Ausstellungsdatum | Unterschrift

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

Einlieferungsschein/Lastschriftzettel

DM | Pf

für Postscheckkonto Nr. 14 199-803

Postscheckamt München

für Markt&Technik Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in 8013 Haar

PROGRAMM-SERVICE

Berechnung realistischer Bilder mit 3D-Effekt

Raytracing: Mit dem Programm des Monats können Sie fantastische Bilder berechnen lassen, wobei der Strahlengang des Lichts berücksichtigt wird. **Termin:** Sie vergessen nie wieder Termine mit diesem Programm, das die Daten der nächsten 15 Tage automatisch anzeigt. **Chain Reaction:** Ein spannendes Spiel, das viel Taktik erfordert und praktisch nur auf dem Computer realisierbar ist. Lösen Sie eine Kettenreaktion aus! **EHB-Demo:** Wie man 64 Farben im Lores-Modus benutzen kann, zeigt dieses C-Programm, das den Extra-Half-Brite-Modus verwendet. **Copper:** Auch von Basic aus kann der Copper für erstaunliche Effekte eingesetzt werden. Die Verwendung der nötigen Bibliotheken können Sie dabei erlernen. **Cursor Cycle:** Diese Utility macht Schluß mit dem unscheinbaren Cursor, indem es dessen Farbe ständig ändert. **MyCLI:** Eine unentbehrliche Hilfe für alle, die CLI-Fenster mit besonderer Voreinstellung benötigen. **MEd:** Das Schreiben von C-Programmen für Pull-down-Menüs verkürzt sich auf wenige Minuten. Mit einem weiteren Teil können Sie dann CLI-Befehle im Pull-down-Menü verwenden.
1 Diskette für Amiga

Bestell-Nr.: 48704 **DM 29,90*** sFr 24,90*/öS 299,–*

Super-Malprogramme

Rainbow-Drawer: Dieses Programm des Monats bietet leistungsfähige Befehle und Funktionen, wie sie von professionellen Programmen bekannt sind: bis zu 32 Farben, alle Auflösungen, viele Befehle zum Zeichnen sowie FILL mit Mustern, BOW und anderem. **Turtle:** Mit dieser Befehlserweiterung verfügen Sie über die Grafikbefehle, die bei Logo bekannt und beliebt sind. **Fractals:** Dreidimensionale, realistische Gebirge mit Schattierung erzeugt dieses Programm. **Clouds:** Genauso wirklichkeitsnah wie die Gebirge, aber noch erstaunlicher, sind die Wolken, die Sie mit Clouds generieren. **Apfelmännchen:** Hiermit erzeugen Sie schöne Grafiken aus der beliebten Mandelbrot-Ebene. **Kudiplo:** Ein gutes, unverzichtbares Werkzeug für die Kurvendiskussion stellt »Kudiplo« dar. **Senso:** Testen Sie mit dieser Computer-Adaption des bekannten Spiels Ihr Gedächtnis! **Division:** Bis zu 32000 Nachkommastellen können durch dieses Programm berechnet werden. **Alert:** Alarme, zum Beispiel die bekannten Guru-Meditations, können Sie nun selbst programmieren. Das Programm ist in erster Linie für C-Programmierer aufschlußreich. **Border:** Lassen Sie den Fensterrahmen des CLI-Fensters einfach verschwinden! **SCD:** Mit diesem Utility können Sie den Pfadnamen in der Titelleiste des Fensters anzeigen.
1 Diskette für Amiga

Bestell-Nr.: 48703 **DM 29,90*** sFr 24,90*/öS 299,–*

Disk-Operation vereinfacht

ExDir: Mit dem Programm des Monats der Ausgabe 8/9 sind alle Schwierigkeiten mit den Subdirectories endlich vorbei. **Hardcopy:** Diese nützliche Routine gestattet es, Teile des Bildschirms auszudrucken. **Bio:** Um Ihre körperliche, seelische und psychische Verfassung zu erfahren, können Sie dieses fantastische Biorhythmusprogramm einsetzen. **Mask:** Dieses Utility hilft Ihnen beim Herstellen komplexer Eingabemasken. **Freeman:** Den momentan verfügbaren Speicherplatz erfahren Sie mit dieser Routine im CLI-Window. **Vokabel:** Zum Lernen von Vokabeln ist dieses Programm ein sehr gutes Werkzeug mit viel Komfort. **SimpleSprite:** Wie man Sprites auf dem Amiga programmiert, sehen Sie in diesem Demoprogramm. **Vier gewinnt:** Wenn Sie gern spielen und strategisches Denken trainieren wollen, ist Vier gewinnt genau das Richtige für Sie. Weiterhin finden Sie auf der Diskette ein nützliches Tool zur Veränderung des CLI-Windows (InitCLI).
1 Diskette für Amiga

Bestell-Nr.: 48702 **DM 29,90*** sFr 24,90*/öS 299,–*

Leckerbissen für C-Programmierer

Netzgrafik: Wie Sie auf dem Amiga dreidimensionale Netzgrafiken erstellen, zeigt dieses Programm in Amiga-Basic. **Disketten-Monitor:** Dieser Monitor, geschrieben in Amiga-Basic, bietet viele Möglichkeiten bei der Diskettenmanipulation. **Menü-Routinen:** Benutzen Sie die Betriebssystemroutinen von C zum Erzeugen von Pull-down-Menüs. **Polydraw:** Wie Sie die Grafik-Bibliothek in Amiga-Basic sinnvoll nutzen, zeigt dieses Programm. **Font-Designer:** Amiga-Basic-Programm, das Ihnen ermöglicht, eigene Zeichensätze herzustellen. **FTYPE:** In C geschriebene Druckroutine mit viel Komfort. **FCat:** Schnelle Directory-Routine. **Objekt-Editor:** Ein Sprite- und BOB-Editor mit komfortabler Maussteuerung.
1 Diskette für Amiga

Bestell-Nr.: 48701 **DM 29,90*** sFr 24,90*/öS 299,–*

Super-Utilities für Grafik, Sound und Disk

Brush-Converter: Ein Programm, das Deluxe-Paint-Brushes in Amiga-Basic-Objekte umsetzt. **Gesellschaftsspiel:** Punktesammeln durch das richtige Beantworten vieler Fragen aus den unterschiedlichsten Bereichen, die Sie selbst eingeben. **Fragen zum Spiel:** 300 Fragen aus den Bereichen Naturwissenschaften und Allgemeinwissen für das Gesellschaftsspiel. **Grafik-Tool:** Grafikspeicherprogramm legt auf Maustastendruck die aktuelle Bildschirmgrafik im IFF-Format auf Diskette ab. **Disk-Utility:** Kopieren, Umbenennen und Löschen von Dateien sowie das Formatieren und Duplizieren von Disketten über eine komfortable Benutzeroberfläche. **Sound-Digitizer:** Schnell und einfach digitalisieren Sie jeden beliebigen Klang mit der Steuersoftware. Viele Funktionen und Effekte stehen dafür zur Verfügung. Mit vielen Demonstrationsklängen zum Experimentieren. Mit dieser Zusatz-Software arbeitet der Sound-Digitizer (Bauanleitung in 68000er Ausgabe 1/87) auch mit dem Amiga 500 und dem Amiga 2000 einwandfrei zusammen. **Blitter-Demo:** Ein kleines Programm zur Demonstration des schnellen Amiga-Blitters. Fliegen Sie nach Belieben in einer dreidimensionalen Vektor-Landschaft umher.
1 Diskette für Amiga

Bestell-Nr.: 33707 **DM 29,90*** sFr 24,90*/öS 299,–*

Endlich: Pixelgenaue Hardcopies

Drucker-Utility: Druckroutine zur pixelgenauen Ausgabe beliebiger IFF-Grafiken auf Epson-kompatiblen Druckern. **Terminalprogramm:** Amiga-Basic-Programm, das alle Grundfunktionen professioneller DFÜ-Software beinhaltet. **Schnelles Kopierprogramm:** Mit diesem C-Programm sind weniger Diskettenwechsel beim Kopieren mit nur einem Laufwerk notwendig. **CLI-Tool:** Die Bedienung des Amiga-CLI erleichtern drei Amiga-DOS-Dateien in der RAM-Disk. **Sprachausgabe:** Zahlen bis zu einer Größe von einer Milliarde spricht dieses Programm perfekt in Deutsch aus. **Betriebssystem-Offset:** Alle Offsets der Amiga-Betriebssystem-Routinen für Besitzer des SEKA-Assemblers.
1 Diskette für Amiga

Bestell-Nr.: 32704 **DM 29,90*** sFr 24,90*/öS 299,–*

Software für den Amiga-Sound-Digitizer

Digitizer: Testprogramm und Sound-Demos zum Amiga-Digitizer. Digitalisieren Sie beliebige Sounds aus dem Radio, von Platte oder Band. Die Beschreibung und die Bauanleitung finden Sie im 68000er-Magazin Ausgabe 1/87. **Senso:** Ein tolles Denkspiel in Amiga-Basic. Ihre Aufgabe ist es, eine Ton- und Farbfolge, die der Computer vorgibt, richtig nachzuspielen. **Stoneage:** Basic-Geschicklichkeitsspiel für Amiga-Computer mit mindestens 512-Kbyte-RAM, bei dem das beliebte Spiel Boulder Dash Pate stand. Begleiten Sie unseren Helden Willi Beisser auf seiner gefährlichen unterirdischen Mission. **Dreidim:** Dieses C-Programm demonstriert die Verwendung der schnellen Fließkommaroutinen (FFP) des Amiga-Betriebssystems anhand dreidimensional dargestellter Funktionen.
1 Diskette für Amiga

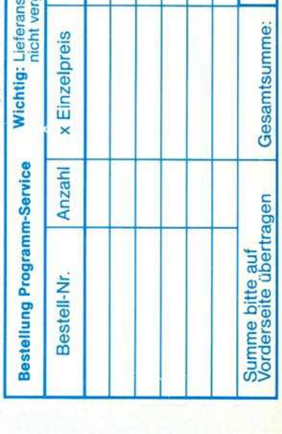
Bestell-Nr.: 32701 **DM 29,90*** sFr 24,90*/öS 299,–*

* Unverbindliche Preisempfehlung

Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides!
Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jede Woche erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.
Wenn Sie Fragen zu den Programmen in unserem Angebot haben, rufen Sie uns an: Telefon (089) 4613-640 oder (089) 4613-133.

Bei Fragen zu Bestellung und Versand der Programmservice-Disketten wählen Sie bitte Telefon (089) 4613-232.
Bestellungen bitte an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656. Österreich: Ueberreuter Media Handels- und Verlagsgesellschaft mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die abgedruckte Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.



Einlieferungsschein/Lastschriftzettel
(nicht zu Mitteilungen an den Empfänger benutzen)
Gebühr für die Zahlkarte
(wird bei der Einlieferung bar erhoben)
bis 10 DM 90 Pf
über 10 DM (unbeschränkt) 1,50 DM
Bei Verwendung als Postüberweisung gebührenfrei

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos
Auskunft hierüber erteilt jedes Postamt

Feld für postdienstliche Zwecke

Hinweis für Postgirokontoinhaber:
Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.
1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgiroteilnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

Abkürzungen für die Ortsnamen der PGiroÄ:
Bln W = Berlin West; Dtmd = Dortmund; Esn = Essen; Ffm = Frankfurt am Main; Hmb = Hamburg; Han = Hannover; Krlh = Karlsruhe; Kln = Köln; Lshfn = Ludwigshafen am Rhein; Mchn = München; Nbg = Nürnberg; Sbr = Saarbrücken; Stgt = Stuttgart

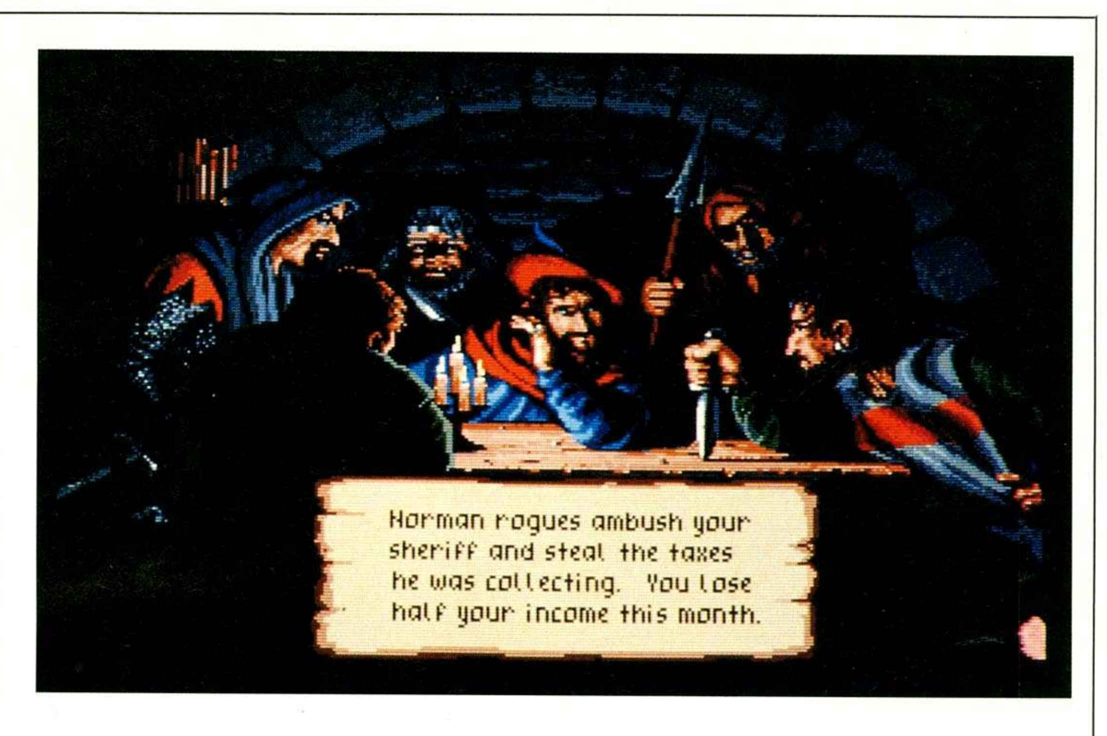
Für Mitteilungen an den Empfänger

Bestellung Programm-Service: Bestell-Nr.	Anzahl	x Einzelpreis	= Gesamtpreis

Wichtig: Lieferanschrift (Rückseite) nicht vergessen!
Summe bitte auf Vorderseite übertragen
Gesamtsumme:

Der große Spieleschwerpunkt

Noch rechtzeitig zum Weihnachtsgeschäft bringen wir die Informationen für heiße Spieleabende unterm Christbaum. Eine große Übersicht stellt Ihnen die besten Programme vor, die der Amiga zur Zeit zu bieten hat. Die Palette reicht von Action- und Sportspielen bis in die Bereiche Strategie und Adventure. Da sollte für jeden etwas Passendes dabeisein. Außerdem gibt es einen erweiterten Teil mit Spieletests. Zusätzlich geben wir Tips unter dem Motto »Worauf man beim Spielekauf achten sollte«, und wir befassen uns damit, wie man am besten an schwierige Probleme bei Abenteuerspielen herangeht und sich so Stunden mühevollen Grübelns erspart.



Alles an Programmiersprachen

Für alle, die gerne programmieren, kommt der Schwerpunkt Computersprachen in unserer nächsten Ausgabe gerade richtig. Da Basic nicht nur für Einsteiger eine interessante Sprache ist, werden wir drei Basic-Dialekte im direkten Vergleich testen. Das bei allen Amiga-Modellen mitgelieferte Amiga-Basic muß gegen True-Basic und SAM-Basic antreten. Wir sind genauso gespannt wie Sie, wer als Sieger aus diesem Kopf-an-Kopf-Rennen hervorgehen wird. Ein weiterer Vergleich dürfte für Fortgeschrittene von großem Interesse sein: Modula 2 gegen C. Die Frage, die es zu beantworten gilt, ist: Kann Modula 2 eine Alternative zu C sein? Außerdem geben wir einen Überblick über die zwei bekanntesten Sprachen der Künstlichen Intelligenz: Prolog und Lisp sind leistungsfähige Sprachen, die speziell für solche Anwendungen konzipiert wurden.



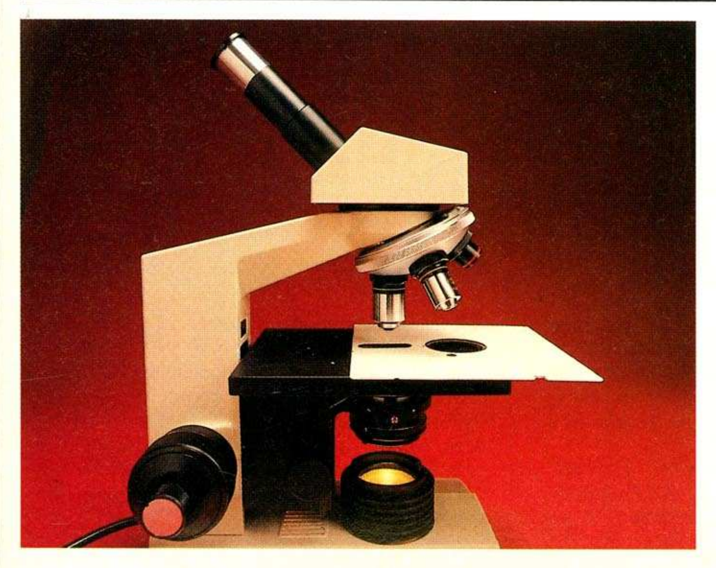
Die neuesten Grafikwerkzeuge

Eine Slide-Show für Bilder, die im IFF-Standard auf Diskette vorliegen, gibt es als Public Domain-Programm schon lange. Wer jedoch Bilder in schneller Reihenfolge durchschalten will, um auf diese Weise eine Quasi-Animation der Zeichnungen vorzunehmen, wird erst jetzt mit dem »Page Flipper« ein geeignetes Werkzeug finden. Sehr gut eignet sich solch ein Tool als Zusatz für Grafikprogramme, die selbst keine Animation vorsehen. Dazu gehört auch der »Landscape Designer«, ein neuer Fractal-Maker, der auch in der nächsten AMIGA getestet wird.

Vorschau 1/88

AUSSERDEM IN DEN NÄCHSTEN AUSGABEN:
MALWETTBEWERB — DIE VORSTELLUNG DER GEWINNER
PROGRAMMIERWETTBEWERB — HERAUSFORDERUNG FÜR ALLE PROGRAMMIERER
SPRACHEN FÜR EINSTEIGER — GROSSE ÜBERSICHT ALLER PROGRAMMIERSPRACHEN
UND WIEDER VIELE HILFEN FÜR JEDEN AMIGA-BESITZER

Die nächste Ausgabe erscheint am 18. 12. 1987. Erhältlich bei Ihrem Zeitschriften-Händler.



Amiga in Quarantäne

Vor Computerviren haben nicht nur Raubkopierer Angst. Auch Käufer von Originalen beschwerten sich schon über das Auftauchen von Viren-Programmen auf frisch gekauften Disketten. Das Einlegen der Disketten in das Laufwerk reicht schon für eine Infektion; das kann auch im Computergeschäft passieren, wenn zur Demonstration mehrere Programme hintereinander geladen werden. Was läßt sich dagegen tun? Dieser Frage werden wir in einer Story in der nächsten AMIGA nachgehen.

Wörter in Perfektion

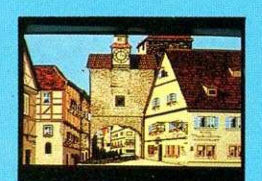
Wirklich gute Textverarbeitungsprogramme, die mit durchdachten Funktionen aus bekannten PC-Programmen glänzen, waren bisher auf dem Amiga nicht besonders viele zu finden. Das soll sich jetzt grundlegend mit dem Erscheinen von »Word Perfect« ändern. Kann Word Perfect den Ansprüchen gerecht werden, zumal es doch zu einem relativ hohen Preis verkauft wird? An Schnelligkeit sowie vielen Funktionen zur Bearbeitung auch großer Texte soll es laut Hersteller nicht mangeln.

Unsere

SPIELE DES JAHRES

für den

COMPUTER DES JAHRES





EMETIC SKIMMER

von Joachim Deeg

Wir schreiben das Jahr 5354. Die Erde hat soeben den ersten Computer mit künstlicher Intelligenz fertiggestellt und hat ihn auf dem fernen Planeten Pluto stationiert. Aber die Wissenschaftler haben die Fähigkeiten des Computers unterschätzt. Mit seinem immensen Wissen und seinem fehlenden Gewissen stellt er eine ernstzunehmende Gefahr für das gesamte Sonnensystem dar. Ihre Aufgabe ist es nun, den Emetic Skimmer, ein Raumschiff mit Spezialausstattung, durch die Verteidigungslinien des Computers zu steuern und ihn am Ende zu zerstören. Das neueste Superspiel für den Amiga - 2 Disketten randvoll gepackt mit Spitzen-Grafik, Digisounds und jeder Menge Action.







MIKE - THE MAGIC DRAGON

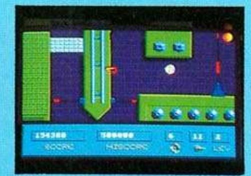
von Frank Neuhaus

Mike ist ein kleiner Drachenjunge, der von dem bösen Professor Dragan Drachenklau entführt worden ist, denn dieser benutzt in seinen Burgen und Schlössern mit Vorliebe kleine Drachen für seine garstigen Experimente. Doch Mike hat sich von den Ketten befreien können und sucht nun einen Weg durch das Labor des Professors, der in die Freiheit führt. Ein Kletter- und Springspiel der Extraklasse mit vielen neuartigen Extras.











PHALANX II - THE RETURN

nach Oliver Czesla

Nach dem Riesenerfolg des Actiongames PHALANX ist der langersehnte Nachfolger jetzt fertig. Und das Warten hat sich gelohnt, denn Grafik, Sound und Action haben eine weitere Steigerung erfahren. 1 oder 2 Spieler (gleichzeitig!) rasen über die vertikal scrollende Landschaft auf der Suche nach immer neuen Formationen der Gegner.



SIE GLAUBEN KEINEN WERBESPRÜCHEN?!? DANN LESEN SIE SELBST, WAS DIE DEUTSCHE FACHPRESSE ÜBER KINGSOFT-SPIELE SCHREIBT:

CHALLENGER von Andreas von Lepel

„Challenger ist kein Spiel, das man zur Spitzenklasse zählen kann, trotzdem ist es wegen seines relativ niedrigen Preises zu empfehlen." (ASM) **29.95 DM**

CITY DEFENCE von Uwe Kühner

„Originalgetreue Grafik, guter Sound, fantastische Steuerung und, was vor allem wichtig ist, eine gute Spielatmosphäre. Da es für den Amiga im Moment außerdem an guten Action-Spielen mangelt, kann man City Defence also nur wärmstens empfehlen, denn es ist wirklich Klasse!" (ASM) **29.95 DM**

CRUNCHER FACTORY von M. Aberle

„Vor allem zu zweit macht Cruncher Factory Spaß und angesichts des sehr attraktiven Preises kann man das Spiel empfehlen." (HAPPY COMPUTER)
„Eine amüsante und schnelle Pacman-Variante ist Cruncher, der zudem noch mit einem Level-Editor versehen ist. ... Die Grafik ist ausreichend gut, die digitalisierte Titelmusik wieder einmal hörenswert. Wer diesen Klassiker gerne für den Amiga haben möchte, sollte die Gelegenheit wahrnehmen." (KICKSTART) **29.95 DM**

DEMOLITION von Martin Aberle

„Von der Grafik her ist das Spiel recht gut, auch der digitalisierte Sound ist hörenswert." (KICKSTART) **29.95 DM**

EMERALD MINE

von Klaus Heinz & Volker Wertich

„Emerald Mine ist zur Zeit eines der besten (wenn nicht sogar das beste!) für den Amiga erhältliche Spiel. Da soll noch mal einer behaupten, daß in Deutschland keine guten Spiele produziert werden. Emerald Mine hebt sich von den bisher vorherrschenden Shoot 'em up Games deutlich ab und setzt in der Amiga-Softwareszene neue Maßstäbe. Dafür sorgt auch der unglaubliche Preis von nur knapp 30.- DM." (AMIGA SPECIAL)
„Die Steuerung des (oder der) Männchen(s) erfolgt mit dem Joystick und ist sehr exakt. So richtig Spaß macht das Game dann, wenn man zu zweit spielt. Zum Programm Emerald Mine kann man Kingsoft nur gratulieren, was einem bei diesem Spiel alles geboten bekommt, das sucht seinesgleichen. Nimmt man noch den überaus günstigen Preis von ca. 30 DM, dann ist mit Emerald Mine ein richtiger Hit entstanden, der jede Spielesammlung bereichert." (ASM)
„Die flotte Grafik mit witzigen Monster-Sprites und vor allem der Sound gefallen mit sehr gut: Die fetzige Titelmusik und die digitalisierten Sound-Effekte sind exzellent. Das Verblüffendste an diesem tollen Spiel ist aber der sehr niedrige Preis." (HAPPY COMPUTER)
„Grafik und Scrolling dieses Spieles sind überzeugend. ... Insgesamt hinterließ Emerald Mine einen guten Eindruck." (KICKSTART) **29.95 DM**

FLIP FLOP Amiga-Version von Edgar Törnig

„Das Programm ist für meinen Geschmack fast schon zu gut. Ich will ja nicht von mir behaupten, daß ich gut Reversi spielen kann, aber selten zuvor bin ich mit solch fliegenden Fahnen untergegangen." (ASM) **29.95 DM**

FORTRESS UNDERGROUND

von Andreas von Lepel

„Die Steuerung des Hubschraubers ist sehr exakt, in dieser Hinsicht ist das Programm gelungen. Auch das Scrolling geht in Ordnung. Der Sound kann sich hören lassen." (ASM) **29.95 DM**

KARATE KING von Marc Ebner

„Im Vergleich zu den beiden anderen Karatespielen ist Karate King das schnellste." (KICKSTART) **49.95 DM**

PINBALL WIZARD

von Hans Georg Berg

„Computer-Flippern stehe ich eigentlich skeptisch gegenüber, weil der Spielablauf größtenteils vom Zufall abhängt. Pinball Wizard ist aber so clever gemacht, daß er mich zu längeren Flipper-Sessions verführen kann. Die schön gezeichnete Grafik und die Digi-Sounds können überzeugen." (HAPPY COMPUTER)
„Die Idee, auf einem Homecomputer einen Flipper zu programmieren, ist zwar nicht neu, aber ich glaube, daß dem Programmierer von Pinball Wizard ein großer Wurf gelungen ist. ... Pinball Wizard ist die realistischste Umsetzung dieses Spiels, die ich kenne. ... Die Grafik, die dem Spieler geboten wird, ist vom Feinsten. ... Auch nach längerer Spieldauer konnte ich keine Schwachpunkte entdecken, so daß ich dieses Programm unbedingt empfehlen kann. Für mich ist Pinball Wizard ein Spiel, das in keiner (Amiga-) Sammlung fehlen darf." (ASM) **49.95 DM**

QUIWI Amiga-Version von Andreas Arens

„Fazit: ein sehr gutes Computer-Gesellschaftsspiel mit Zukunft." (HAPPY COMPUTER)
„Sogar die zehnte Revancherunde macht noch Spaß, denn 4000 Fragen machen Wiederholungen selten. QUIWI ist ein amüsantes Quizspiel für Feste und Familienfeiern." (RUN)
„QUIWI hingegen ist ein wirklich spaßiges wie unterhaltsames und lehrreiches Programm, wenn man mir diese Wiederholung von vorher verzeihen mag. Empfehlenswert!" (ASM)
„Ein abwechslungsreiches Spiel für die ganze Familie." (CHIP) **49.95 DM**

SOCCER KING

von Wolfgang & Helmut Theis

Fußballspiel für 1/2 Spieler; noch nicht getestet. **49.95 DM**

SPACE BATTLE von Andreas Hommel

„Space Battle ist ein solide gemachtes Spiel, das trotz der alten Spielidee viel Spaß bringt. Wenn man bedenkt, daß sowohl Cruncher Factory als auch Space Battle unter 30 Mark kosten, dann fällt einem die Entscheidung zum Kauf nicht mehr schwer. ... Beide haben, obwohl sie billig sind, mehr zu bieten als ähnliche Spiele." (AMIGA-MAGAZIN)
„Für knapp dreißig Mark erhält man mit Space Battle zwar kein neuartiges Spiel. Trotzdem ist es, besonders im Competition-Modus, spannend." (KICKSTART) **29.95 DM**

TYPHOON

von Christoph Sing & Rolf Wagner

„Auf diesem Hintergrund tummeln sich Massen von Sprites, die hervorragend ausgearbeitet und animiert sind. Es macht wirklich Laune, die verschiedenen Varianten von Raumschiffen und Hindernissen zu betrachten. ... Trotz einiger taktischer Varianten ist TYPHOON ein reinrassiges Actionspiel, das die Fans dieses Genres begeistern dürfte. Besonders die 50 verschiedenen Screens mit der erstklassigen Grafik und den blitzsauber animierten Sprites garantiert Freunden des Actionspiels lange Spielmotivation. ... Tolle Grafik und viel Action in TYPHOON ergänzen sich zu einem unterhaltsamen Spiel." (68000er)
„Wenn mich zukünftig jemand fragt, was ich unter einem guten Action-Spiel verstehe, werde ich ihn an TYPHOON verweisen. Die Handlung ist zwar dürftig bis nicht vorhanden, aber die Grafik ist ungeheuer faszinierend. ... Die Asteroiden-Felder, der Dschungel und die Stadt haben mich sofort begeistert. Allein die Neugierde, welche Sprites in der nächsten Angriffswelle kommen, hat mich vor den Monitor gebannt. ... Die Grafik steht so manchem Spielautomaten in nichts nach. ... Das Scrolling ist samtweich, Sprites sind reichlich vorhanden und gut animiert. Der Schwierigkeitsgrad des Spiels steigert sich ganz allmählich und die Idee mit den drei Extra-Waffen weist die Monotonie in die Schranken, die sonst nach einer Weile bei Ballerspielen entsteht. Die Präsentation kann sich auch sehen lassen: Schickes Titelbild, digitalisierte Disco-Rhythmen und eine prächtige High-Score-Liste." (HAPPY COMPUTER) **49.95 DM**

WILLY THE KID von Peter Meyer

Deutsches Text-/Grafik-Adventure; noch nicht getestet. **29.95 DM**



Bei uns gibt's jeden Monat Neuerscheinungen für Ihren AMIGA!

SPITZEN-SOFTWARE



MADE IN GERMANY

Alle Spiele selbstverständlich mit ausführlicher deutscher Anleitung, Super-Grafik und fantastischen Digisound-Effekten. Grauimporte haben keine deutsche Anleitung!

F. SCHÄFER · Schnackebusch 4
D-5106 ROETGEN ☎ 02408/5119
(nicht aufgeben!) Fax 02408/5213

KINGSOFT-Coupon

☐ Bitte senden Sie mir Ihren aktuellen Katalog

Meine Adresse

COMMODORE AMIGA 500 – FÜR ALLE, DIE IMMER VORNE SIND

